FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2017/2018

Trabajo de Fin de Máster

TITULO: **Construcción de un modelo de predicción para la puntualidad de vuelos comerciales.**

***Alumno*: Iván Esperón Cespón**

***Tutora*: Aida Calviño Martínez**

Junio de 2018

UNIVERSIDAD COMPLUTENSE
MADRID

A mis padres y a mis abuelos.

Agradecimientos

En primer lugar agradecerle a Aida Calviño Martínez su ayuda, consejo y dedicación para la realización de este Trabajo Fin de Máster, muchas gracias por la confianza depositada en mí durante todo el máster.

A mis padres, por apoyarme en todo momento y empujarme a hacer este máster que me ayudará en mi futuro, gracias por el esfuerzo y la confianza. A mis abuelos porque siempre me han ayudado y aconsejado con lo que han creído que es lo mejor para mí. A mi hermano porque es capaz de sacarme una sonrisa en cualquier momento y situación. A mi familia en general por apoyarme siempre en todo. Y a Marina, porque sin ti todo hubiese sido más difícil, gracias por estar a mi lado en todo momento.

Muchas gracias a todos.

Resumen

En base al crecimiento continuo de los últimos años en el transporte aéreo de pasajeros y con la finalidad de estudiar los motivos que producen retrasos en los vuelos, se propone este TFM.

Para ello se va a desarrollar un cuadro de mando con el que poder visualizar de forma sencilla y clara toda la información con la que se trabaje y un modelo predictivo que ayude a entender y predecir si un vuelo comercial tendrá o no retraso y el tiempo que conllevará el mismo.

Para llevar a cabo esta tarea, se utilizan datos del año 2016 de los 6 aeropuertos con más tránsito de pasajeros de Estados Unidos:

- Hartsfield-Jackson Atlanta International Airport. Atlanta.
- Los Angeles International Airport. Los Ángeles.
- O'Hare International Airport. Chicago.
- Dallas/Ft Worth International Airport. Dallas.
- John F. Kennedy International Airport. Nueva York.
- Denver International Airport. Denver.

Palabras clave

Vuelos, Modelos predictivos, Control de puntualidad, SAS, R, Qlik Sense, Redes Neuronales, Árboles de decisión, Inteligencia de Negocios, Machine Learning.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En el primer capítulo de esta memoria se tratan los aspectos básicos del trabajo, el contexto del Trabajo Fin de Máster (TFM) y la estructura de esta memoria.

1.1. Contexto

Analizando los datos del International Air Transport Association (IATA), que publica en su comunicado Nº 5 del 1 de Febrero de 2018 [1], como vemos en la Figura 1.1, la demanda Revenue Passenger Kilometres (RPK) en el 2017 aumentó un 7,6 % con respecto al 2016 y superó la tasa promedio de crecimiento de la última década que se sitúa en el 5,5 %.

Resultado detallado del mercado aéreo de pasajeros – Diciembre, 2017

	Cuota mundial [1]	Diciembre 2017 (% interanual) RPK	ASK	PLF (%-pt)[2]	PLF (nivel)[3]	Calendario anual 2017 (% interanual) RPK	ASK	PLF (%-pt)[2]	PLF (nivel)[3]
Total Mercado	**100,0%**	**6,2%**	**5,8%**	**0,3%**	**80,7%**	**7,6%**	**6,3%**	**0,9%**	**81,4%**
África	2,2%	3,4%	2,6%	0,6%	72,1%	6,3%	2,9%	2,3%	70,9%
Asia-Pacífico	33,7%	9,1%	8,3%	0,6%	81,1%	10,1%	8,4%	1,3%	81,0%
Europa	26,5%	6,1%	4,4%	1,3%	81,5%	8,2%	6,2%	1,5%	83,9%
Latinoamérica	5,2%	5,4%	5,0%	0,3%	81,5%	7,0%	5,5%	1,2%	81,8%
Oriente Medio	9,5%	3,4%	5,7%	-1,7%	75,5%	6,4%	6,5%	-0,1%	74,5%
Norteamérica	23,0%	4,0%	4,2%	-0,2%	82,7%	4,2%	4,1%	0,1%	83,6%

[1]% RPK de la industria en 2017; [2]Variación interanual del factor de ocupación; [3]Nivel del factor de ocupación.

Figura 1.1: RPK de la industria en 2017.

En Norteamérica la demanda de pasajeros aéreos internacionales registró la mayor aceleración desde 2011. Los RPK aumentaron un 4.8 %.

En 2011 la Comisión Europea publicaba un informe [2] donde se predice que en 2030 diecinueve aeropuertos europeos estarán saturados por la congestión del tráfico aéreo, lo que podría provocar retrasos que afectarán al 50 % de los vuelos de pasajeros y mercancías.

Con estos datos relativos al crecimiento de pasajeros y en previsión de que se pueda llegar a estados de congestión en diversos aeropuertos se plantea este trabajo fin de máster. Para ello se estudiará el tráfico aéreo de los 6 aeropuertos más importantes de Estados Unidos y se buscará predecir el tiempo de retraso que producirá una ruta aérea según diversas condiciones.

Para la visualización de los datos relativos al tráfico aéreo se creará un cuadro de mando en el que poder explotar todas las fuentes de datos involucradas en este trabajo.

1.2. Estructura de la memoria

Esta memoria consta de 6 capítulos, 2 apéndices, glosario, lista de acrónimos y bibliografía, en donde se explica la realización de este trabajo.

Capítulo 1 - Introducción Se da una visión general del contexto del trabajo y la estructura de la memoria.

Capítulo 2 - Fuentes de datos Se muestran las diferentes fuentes de datos que intervienen en el trabajo y se explica su contenido.

Capítulo 3 - Objetivos y metodología En este capítulo se indican los objetivos del trabajo y se presenta la metodología empleada, SEMMA.

Capítulo 4 - Variables Se definen las variables implicadas en el trabajo y se hace un análisis descriptivo de las mismas.

Capítulo 5 - Modelado Se muestran los diferentes modelos predictivos generados.

Capítulo 6 - Conclusiones y líneas futuras Se describen los objetivos conseguidos y las líneas de trabajo futuras.

Apéndice A - Manual de usuario Se muestran las indicaciones para realizar la

instalación de la herramienta y las instrucciones de uso.

Apéndice B - Código Contiene parte del código de programación empleado en este trabajo.

Capítulo 2

Fuentes de datos

En este capítulo se presentan las diferentes fuentes de datos que se utilizan para la explotación de información y búsqueda del mejor modelo predictivo.

2.1. Naturaleza de los datos

Para este trabajo se ha tenido en cuenta datos desde el 1 de Enero de 2016 hasta el 31 de Diciembre de 2016, teniendo así un año completo para el estudio.

El conjunto de datos que contiene información sobre las diferentes rutas aéreas se extrae de la Bureau of Transportation Statistics (BTS) [1] que es la oficina de estadística de transporte de Estados Unidos y es la principal fuente de estadísticas sobre la aviación comercial, de carga y la economía del transporte.

La información que se puede extraer es diaria en archivos mensuales por lo que se obtendrán 12 archivos en csv con la información de los diferentes vuelos realizados en Estados Unidos en el año 2016.

El conjunto de datos que se refiere a rutas aéreas está compuesto por 35 variables y 5617660 observaciones. Esta información se refiere a rutas que tienen aeropuerto de origen y destino en Estados Unidos (sólo vuelos nacionales) y que son operadas por compañías Estadounidenses, en total 12 compañías.

[1] BTS webpage: `https://www.bts.dot.gov/`.

Al tener un conjunto de datos tan grande se decide limitarlo cogiendo como origen los 6 aeropuertos con más trafico de Estados Unidos independientemente del destino. Con esto el conjunto de datos se reduce a 1361376 observaciones.

La variable relativa a la aerolínea que opera la ruta viene su código OACI, cuya correspondencia en nombre se extrae de un proyecto en GitHub [2] del usuario bbejeck.

De la web `http://ourairports.com/` se extrae información referente a los aeropuertos. Son un total de 52804 aeropuertos y la variable más interesante de este conjunto de datos es la clasificación de los aeropuertos, donde tenemos los siguientes niveles:

- *close*: Aeropuertos inutilizados.
- *heliport*: Helipuertos, sólo está permitido que operen helicópteros.
- *small_airport*: Más de 10 operaciones de vuelos y más de 15 aviones con base.
- *medium_airport*: Más de 1000 operaciones de vuelos, 1 base o más de jets, más de 10 vuelos internacionales.
- *large_airport*: Más de 5000 operaciones vuelos, más de 11 bases de jets, más de 20 vuelos internacionales o más de 500 salidas interestatales.

Un factor que podría influir en el retraso de los vuelos comerciales son las condiciones climatológicas. Por esto se buscan datos del año 2016 relativos a las 6 ciudades de origen. Esta información está en la National Oceanic and Atmospheric Administration (NOAA) [3] que es una agencia científica del Departamento de Comercio de los Estados Unidos cuyas actividades se centran en las condiciones de los océanos y la atmósfera. De su web se extraen los archivos csv que contienen la información meteorológica. Son 6 archivos, uno por ciudad con 366 observaciones cada archivo.

Por último se extrae un archivo con los días festivos en Estados Unidos desde el 2012 hasta 2020. Este archivo está en un proyecto GitHub del usuario shivaas que se encuentra en la web `https://gist.github.com/shivaas/4758439`.

[2] Spark-experiments project webpage: `https://github.com/bbejeck/spark-experiments/blob/master/src/main/resources/airtraffic/L_AIRLINE_ID.csv-`.

[3] NOAA webpage: `http://www.noaa.gov/`.

2.2. Extracción, Transformación y Carga

Se hace un proceso de Extract, Transform and Load (ETL) con Qlik Sense. Las fuentes de datos descritas en la sección 2.1 componen el conjunto de datos que se utilizará en este TFM.

Para extraer la información de ellos, explotarla y por último generar un único archivo que contiene todos los datos, se utiliza la herramienta Qlik Sense Desktop [4] que ofrece a los individuos la posibilidad de crear visualizaciones de datos, informes y cuadros de mando personalizados e interactivos a partir de múltiples fuentes de datos.

La **extracción** se hace desde el script de carga de Qlik Sense. Para mejorar los tiempos de carga de los datos en la aplicación se genera un QlikView Data (QVD) que es un formato nativo de Qlik. El formato de archivo está optimizado para mejorar la velocidad de lectura de datos desde un script, siendo al mismo tiempo muy compacto. Leer datos desde un archivo QVD es por lo general 10-100 veces más rápido que leer desde otras fuentes de datos. Este QVD contiene todos los datos extraídos de la web `https://www.bts.dot.gov/`, con esto nos ahorraremos utilizar un bucle para cargar los 12 archivos que contienen estos datos.

La **transformación** también se realiza desde el script de carga, generando nuevas variables como pueden ser el tiempo de retraso en la salida del vuelo en minutos (Hora de salida real - Hora de salida), el coeficiente de retraso (Tiempo de retraso/Tiempo de vuelo), una variable que se llamará TipoDía en la que se crean 6 categorías (-2F, -1F, F, +1F, +2F y L) para indicar desde 2 días antes hasta 2 días después los días que son festivos, etc.

Se hacen transformaciones en los campos para cambiarles el formato y así poder utilizarlos en las visualizaciones de la aplicación, como por ejemplo con la latitud y longitud de los aeropuertos. En el siguiente código 2.1 se muestra un ejemplo de una sentencia del script de carga cuya finalidad es traer la información de los aeropuertos de destino transformando sus coordenadas y marcándolas como geolocalizaciones para poder utilizar

[4] Qlik Sense webpage: `https://www.qlik.com/es-es/products/qlik-sense/desktop`.

mapas dentro de la aplicación.

```
LOAD
    type as [Tipo aeropuerto],
    GeoMakePoint(Replace(Text(latitude_deg),'.',','), Replace(Text(
       longitude_deg),'.',',')) as [Localizacion Destino],
    Replace(Text(latitude_deg),'.',',') as [Latitud destino],
    Replace(Text(longitude_deg),'.',',') as [Longitud destino],
    municipality as [Ciudad destino],
    iata_code as [Destino]
FROM
['lib://Aeropuertos\airports.csv']
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
where type<>'closed';
```

Código 2.1: Carga de información de los aeropuertos destino.

Con todo esto se crea un modelo de datos asociativo dónde toda la información estará relacionada, como se muestra en la Figura 2.1 donde tenemos dos tablas de hechos que son *Vuelos y Meteorología*, dos tablas de dimensión *Calendario y AeropuertosOrigen* y una tabla de enlace entre las dimensiones y las tablas de hechos.

Todos estos datos se **cargan** en una aplicación Qlik Sense para poder visualizarlos y explotarlos.

Por último se genera un archivo csv con toda la información recopilada que se analizará en busca del mejor modelo predictivo.

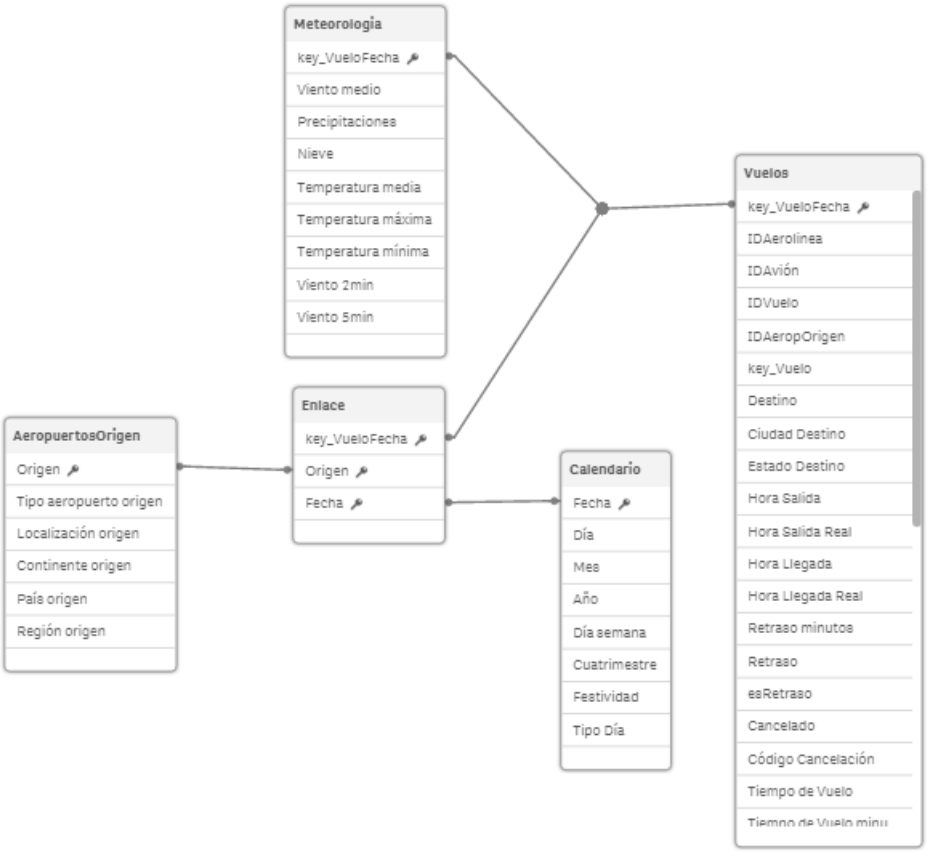


Figura 2.1: Modelo asociativo de la aplicación.

Capítulo 3

Objetivos y metodología

Se definen los objetivos de este TFM y se describe la metodología Sample, Explore, Modify, Model, and Assess (SEMMA)

3.1. Objetivos

El objetivo principal de este trabajo es desarrollar un cuadro de mando para la visualización y explotación de los datos de los 6 aeropuertos con más tráfico de Estados Unidos y la búsqueda de un modelo predictivo que nos ayude a adelantarnos a futuras situaciones relativas al tráfico aéreo.

Como objetivos más concretos se marcaron:

- Definir y estudiar las diversas fuentes de datos.
- Visualizar todos los datos relativos a los aeropuertos que se van a estudiar.
- Explotar la información con diferentes gráficos.
- Crear un modelo asociativo que ayude a la visualización y explotación de los datos.
- Definir diferentes métricas que se consideren importantes para la explotación de la información.

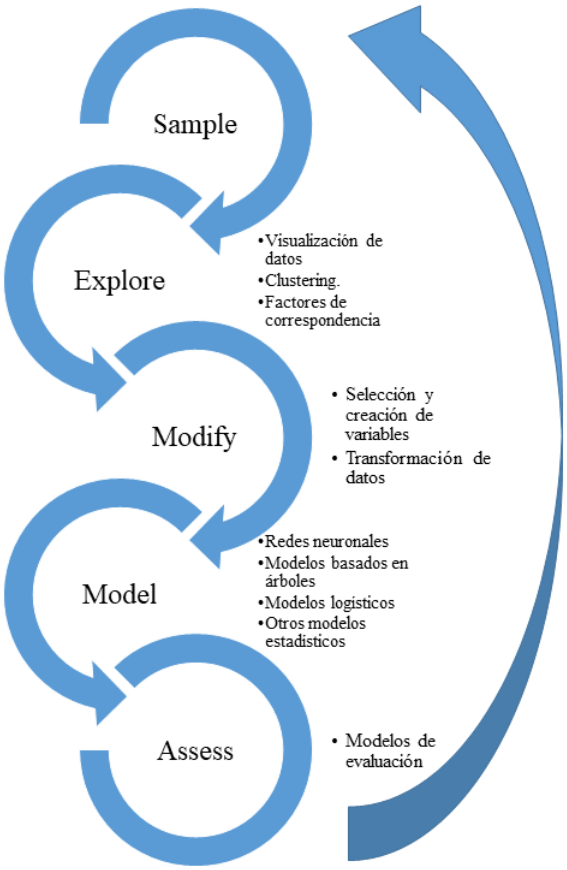


Figura 3.1: Metodología SEMMA.

- Creación de modelos predictivos con el fin de proporcionar un modelo preciso de predicción del tiempo de retraso de un vuelo.

3.2. SEMMA

SAS Institute es el desarrollador de la metodología SEMMA [3], la define como el proceso de selección, exploración y modelado de grandes cantidades de datos para descubrir patrones de negocio desconocidos. El nombre de esta terminología es el acrónimo correspondiente a las cinco fases básicas del proceso que se muestran en la Figura 3.1.

El proceso se inicia con la extracción de la población muestral sobre la que se va a aplicar el análisis. El objetivo de esta fase consiste en seleccionar una muestra representativa

del problema en estudio. Una vez determinada una muestra de la población en estudio, la metodología SEMMA indica que se debe proceder a una exploración de la información disponible con el fin de simplificar en lo posible el problema para optimizar la eficiencia del modelo. La tercera fase de la metodología consiste en la manipulación de los datos, en base a la exploración realizada, de forma que se definan y tengan el formato adecuado los datos que serán introducidos en el modelo. Una vez que se han definido las entradas del modelo, con el formato adecuado para la aplicación de la técnica de modelado, se procede al análisis y modelado de los datos. Finalmente, la última fase del proceso consiste en la valoración de los resultados mediante el análisis de bondad del modelo o modelos, contrastado con otros métodos estadísticos o con nuevas poblaciones muestrales.

La Figura 3.1 de la metodología SEMMA puede sufrir diferentes variaciones, tanto en el orden de la aplicación de la metodología ya que hay procesos que se podrían repetir muchas veces, pasando de unas fases a otras sin respetar el orden como en el contenido, puesto que no siempre intervienen todas las fases en la metodología.

3.2.1. Técnicas de modelado

En esta sección se mostrarán las técnicas de modelado que se utilizarán en este TFM exponiendo sus características principales.

Regresión lineal

Los modelos de regresión lineal [4] son simples y a menudo proporcionan una descripción adecuada e interpretable de cómo las entradas afectan la salida. Con fines de predicción, a veces pueden superar a los modelos no lineales más sofisticados, especialmente en situaciones con un número pequeño de casos de entrenamiento, baja relación señal-ruido o datos dispersos. A los métodos lineales se pueden aplicar transformaciones de las entradas lo que amplía considerablemente su alcance. Tenemos un vector de entrada $XT = (X_1, X_2, ..., X_p)$, y se quiere predecir una salida de valor real Y. El modelo de

regresión lineal tiene la forma

$$Y_t = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \varepsilon \tag{3.1}$$

donde

- Y_t es la variable dependiente u objetivo.
- X_j son las variables explicativas, independientes o de entrada.
- β_i son los parámetros respectivos a cada variable independiente que miden la influencia que las variables explicativas tienen, siendo β_0 el término constante.
- ε es una variable aleatoria, que recoge el error cometido con el modelo. Frecuentemente se asume que su distribución es $N(0, \sigma)$

El modelo lineal supone que la función de regresión E(Y—X) es lineal o que el modelo lineal es una aproximación razonable. Aquí los β_j son parámetros o coeficientes desconocidos, y las variables X_j pueden provenir de diferentes fuentes:

- Entradas cuantitativas.
- Transformaciones de entradas cuantitativas, como log, raíz cuadrada o cuadrado.
- Expansiones de base, como $X_2 = X_1^2, X_3 = X_1^3$, lo que lleva a representación polinómica.
- Codificación numérica o "dummy"de los niveles de entradas cualitativos. Por ejemplo, si G es una entrada de factor de cinco niveles, podríamos crear $X_j, j = 1, ..., 5$, tal que $X_j = I(G = j)$. En conjunto, este grupo de X_j representa el efecto de G por un conjunto de constantes dependientes del nivel, ya que en $\sum_{j=1}^{5} X_j \beta_j$, uno de los X_j es uno, y los otros son cero.
- Interacciones entre variables, por ejemplo, $X_3 = X_1 X_2$.

Dentro de las **ventajas** de este modelo tenemos:

- El análisis de regresión es una herramienta muy flexible en cuanto a la naturaleza de las variables explicativas, pues éstas pueden ser numéricas y categóricas.
- Permite hacer una predicción del comportamiento de alguna variable en un determinado punto o momento.

Sus principales **desventajas**:

- Por su naturaleza, la regresión lineal sólo se basa en las relaciones lineales entre las variables dependientes e independientes.
- La regresión lineal mira a una relación entre la media de la variable dependiente y las variables independientes. Al igual que la media no es una descripción completa de una sola variable, la regresión lineal no es una descripción completa de las relaciones entre variables. Puede hacer frente a este problema mediante el uso de regresión por cuantiles.
- Sensible a los valores atípicos, los valores extremos pueden ser univariado (basado en una variable) o con varias. Los valores atípicos pueden tener enormes efectos en la regresión.
- La regresión lineal asume que los datos son independientes.

Redes neuronales

Una red neuronal artificial [5] es un sistema de procesamiento de información que intenta emular el comportamiento con las redes neuronales biológicas. Las redes neuronales artificiales han sido desarrolladas como generalizaciones de modelos matemáticos del conocimiento humano o de la biología neuronal, con base en las siguientes consideraciones:

1. El procesamiento de información se realiza en muchos elementos simples llamados neuronas.
2. Las señales son pasadas entre neuronas a través de enlaces de conexión.

3. Cada enlace de conexión tiene un peso asociado, el cual, en una red neuronal típica, multiplica la señal transmitida.

4. Cada neurona aplica una función de activación (usualmente no lineal) a las entradas de la red (suma de las señales de entrada pesadas) para determinar su señal de salida.

La distribución de las neuronas dentro de una red neuronal artificial se realiza formando niveles de un número de neuronas determinado. Si un conjunto de neuronas artificiales reciben simultáneamente el mismo tipo de información, lo denominaremos capa. En una red podemos diferenciar tres tipos de niveles:

- **Entrada:** Es el conjunto de neuronas que recibe directamente la información proveniente de las fuentes externas de la red.

- **Oculto:** Corresponde a un conjunto de neuronas internas a la red y no tiene contacto directo con el exterior. El número de niveles ocultos puede estar entre cero y un número elevado. En general las neuronas de cada nivel oculto comparten el mismo tipo de información, por lo que formalmente se denominan Capas Ocultas. Las neuronas de las capas ocultas pueden estar interconectadas de diferentes maneras, lo que determina, junto con su número, las distintas arquitecturas de redes neuronales.

- **Salida:** Es el conjunto de neuronas que transfieren la información que la red ha procesado hacia el exterior.

En la Figura 3.2, se puede apreciar la estructura de capas de una red neuronal artificial con varios niveles.

La capa de entrada se conecta con la capa oculta (θ_j) mediante una función de combinación, donde los pesos W_{ij} (pesos sinápticos) hacen el papel de parámetros a estimar. Sobre esta función se aplica una función de activación, que puede ser, entre otras: función sigmoidal, función gaussiana, función tangente hiperbólica, etc. De los nodos ocultos a los nodos de salida θ_k' se aplica el mismo procedimiento sobre las nuevas variables provenientes de los nodos ocultos: una función de combinación y ocasionalmente una de activación. El valor final de la función de activación en cada nodo oculto es el valor de salida en ese nodo.

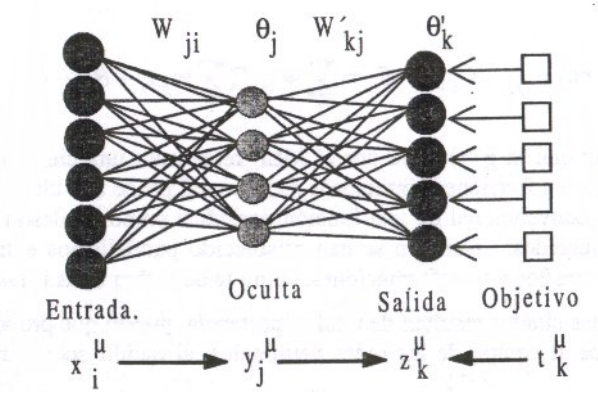


Figura 3.2: Estructura de una red multinivel con todas las conexiones hacia adelante.

Las Redes Neuronales tienen numerosas **ventajas** frente a otras técnicas de predicción, describiendo las más destacables a continuación:

- El aprendizaje del modelo no necesita ser programado, las redes neuronales son capaces de extraer sus propias reglas a partir de ejemplos reales mediante la adaptación de la matriz de ponderaciones. Estas reglas quedan almacenadas y extendidas a lo largo de las conexiones.

- Son tolerantes al ruido, es decir, son capaces de abstraer las características esenciales de los datos y así generalizar de forma correcta aún en presencia de datos distorsionados o incompletos.

- No son paramétricas, no necesitan hacer supuestos de la forma funcional de la función que van a aproximar, ni sobre la distribución de las variables independientes.

- No tienen por qué ser lineales, permiten realizar a través de sus funciones de activación todo tipo de transformaciones de los datos, lo cual supone una gran ventaja frente a los modelos tradicionales de regresión.

Pero tiene dos grandes **limitaciones**:

- La imposibilidad de determinar cómo se procesa internamente la información.

- No existe aún una metodología clara y rigurosa para determinar el número de capas ocultas o el número de nodos que tiene que tener cada capa, lo que hace difícil encontrar el modelo óptimo a la primera, se trata más bien de un proceso de ensayo-error del investigador.

Árboles de decisión

Los árboles de clasificación y regresión [6] constituyen una herramienta útil para la predicción de variables de clase y de intervalo, respectivamente, de una manera sencilla y sin asunciones teóricas sobre los datos. Los árboles representan una segmentación de los datos a partir de una serie de reglas simples, que se van aplicando de forma jerárquica y secuencial. De esta forma, se obtienen una serie de segmentos (llamados nodos) que contienen subconjuntos de la muestra. El segmento original contiene a la totalidad de los datos y recibe el nombre de nodo raíz. Una vez obtenida la segmentación "óptima", entendiendo así aquella que da lugar a nodos con comportamiento homogéneo respecto a la variable objetivo y heterogéneos entre ellos, se asigna un valor de predicción a aquellos nodos que no tienen sucesores (y que reciben el nombre de hojas) de forma que todas las observaciones pertenecientes a dicha hoja serán predichos a partir de dicho valor. Los árboles de decisión se componen por los siguientes elementos:

- **Un nodo raíz** es el punto inicial (y único) del árbol y contiene el conjunto total sobre el que se va a proceder.
- **Nodos padres** es todo aquel nodo que se divide en nodos descendientes.
- **Nodos terminales u hojas** son los nodos que no poseen descendientes, esto es, que no se dividan en otro nivel. Siempre tienen asignada una etiqueta de clase.

Las **ventajas** de este método son las siguientes:

- Los resultados son simples y se comprenden fácilmente.
- Permite encontrar interacciones y reglas difíciles de encontrar con otros métodos.

- Aportan medidas de importancia de las variables.
- Permite tratar los missing de una manera eficiente, que forma parte del proceso de construcción del árbol.
- Son modelos robustos frente a datos atípicos.
- Se pueden incluir costes sobre los errores en las decisiones tomadas.

Sus principales **desventajas** son:

- Cuando la relación entre la variable objetivo y una de entrada es claramente lineal le cuesta modelizarlo.
- Los valores de predicción son "toscos" (mismo valor para todas las observaciones del nodo).
- La construcción de un árbol es computacionalmente compleja.

Las desventajas de los árboles son tratadas con técnicas como Random Forest o Gradient Boosting, por esto, no se realizará un estudio de este método de forma directa y sí usándolos en los algoritmos de Random Forest y Gradient Boosting.

Random forest

Random Forest [7] es un algoritmo de combinación de árboles predictores, tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos basado en árboles de decisión y de clasificación. En este caso, lo vamos a utilizar como árbol de decisión ya que la variable objetivo es de carácter continuo. A continuación, se presenta el esquema de este algoritmo en la Figura 3.3:

Dados los datos de tamaño N.

1. Repetir m veces a), b), c):
   a) Seleccionar nN observaciones con reemplazamiento de los datos originales.
   b) Aplicar un árbol de la siguiente manera:
      - En cada nodo, seleccionar p variables de las k originales y de las p elegidas, escoger la mejor variable para la partición del nodo.
   c) Obtener predicciones para todas las observaciones originales N.
2. Promediar las m predicciones obtenidas en el apartado 1.

Figura 3.3: Algoritmo Random forest.

En general, los parámetros a tener en cuenta en este algoritmo son:

- El tamaño de las muestras, n, y si se va a utilizar bootstrap (con reemplazo) o sin reemplazamiento.
- El número de iteraciones a promediar, m.
- Características del árbol: número de hojas, profundidad, el número de divisiones máximas en cada nodo, el p-valor para las divisiones en cada nodo, y el número de observaciones mínimas en cada rama-nodo.
- Número de variables a muestrear en cada nodo, p.

Este algoritmo incorpora dos fuentes de variabilidad (remuestreo de observaciones y de variables) para mejorar la capacidad de generalización, y reducir el sobreajuste, conservando en cualquier caso la facultad de ajustar bien las relaciones particulares de los datos (interacciones, no linealidad, cortes, problemas de extrapolación, etc.). Las principales **ventajas** de esta técnica son:

- Aumenta la capacidad predictiva y disminuye la varianza.
- Disminuye la sensibilidad frente a cambios en los datos, aumenta la estabilidad y la robustez.

- Aumenta la suavidad (función de predicción menos escalonada), lo que a veces redunda en menor error promedio de predicción.

Por otro lado, esta técnica de predicción presenta la **desventaja** de la pérdida de interpretabilidad de los resultados, donde solo se puede evaluar la importancia de cada una de las variables explicativas del modelo, creándose un ranking de las variables según su frecuencia utilizada en el algoritmo.

Gradient boosting

Gradient Boosting [8] es una evolución de los modelos de Random Forest.

Para poder calcular estos modelos es necesario una primera etapa, en la cual se obtiene el modelo inicial. Este modelo será ajustado en una segunda etapa, ya que la construcción del segundo modelo tendrá en cuenta la información del modelo anterior. El proceso de ajuste modificará las predicciones del modelo anterior con el objetivo de minimizar los errores de los modelos, obteniendo así, de manera gradiente modelos que convergen en un modelo final donde los errores son mínimos.

Ya que la base son los modelos de Random Forest, compartirán las variables que los definen y a su vez ganan nuevos parámetros, los cuales son:

- Iteraciones: reflejan el número de etapas del modelo.
- Parámetro Shrinkage, refleja el grado con el que se ajustará el modelo en cada una de las iteraciones.

Las **ventajas** del Gradient Boosting:

- Invariante frente a transformaciones monótonas: no es necesario realizar transformaciones logarítmicas, etc.
- Buen tratamiento de missing, variables categóricas, etc. Universalidad.
- Muy fácil de implementar, relativamente pocos parámetros a monitorizar (número de hojas o profundidad del árbol, tamaño final de hojas, parámetro de regularización...).

- Gran eficacia predictiva, algoritmo muy competitivo. Supera a menudo al algoritmo Random Forest.
- Robusto respecto a variables irrelevantes. Robusto respecto a colinealidad. Detecta interacciones ocultas.

Su principal **desventaja** es que en datos relativamente sencillos (pocas variables, no missing, no interacciones, linealidad (regresión) o separabilidad lineal (clasificación)), el gradient boosting no tiene nada nuevo que aportar y pueden ser preferibles modelos sencillos (regresión, regresión logística, discriminante) o modelos ad-hoc que adapten aspectos concretos como la no linealidad (redes por ejemplo).

Ensamblado de modelos

Los métodos Ensamble [9] consisten en la construcción de predicciones a partir de la combinación de varios modelos. Existen infinidad de métodos para la combinación de las distintas predicciones. Con el objetivo de mejorar la precisión alcanzada por los modelos de clasificación empleados en el estudio y reducir la varianza de los errores cometidos, se proponen distintos métodos de ensamble de clasificadores mediante la técnica de stacking. Este método consiste en construir clasificadores dados por la combinación, lineal o no, de las probabilidades estimadas por los modelos ajustados, algunos de los cuales son ensambles en sí mismos (Random Forest, Gradient Boosting). Con ello se consiguen las probabilidades estimadas conjuntas y se realiza la clasificación mediante la técnica del punto de corte óptimo de la probabilidad estimada.

Las principales **ventajas** del ensamble de modelos son:

- Bastante robustos, unos modelos corrigen a otros.
- Reducen la varianza del error en general, casi nunca empeoran los modelos.

Por otro lado se comentan las principales **desventajas** de los métodos ensamble:

- Cada modelo tiene sus errores de estimadores de parámetros lo que aumenta aparentemente la complejidad.

- Excesivas posibilidades que a veces llevan al sobreajuste.
- Los resultados no son interpretables.

3.2.2. Comparación de modelos

Una vez aplicadas todas las técnicas mencionadas en la sección anterior 3.2.1, nuestra finalidad es escoger cuál de esas técnicas se ajusta mejor a nuestros datos, es decir, elegiremos el mejor modelo. Para ello, nos vamos a basar en el Error Cuadrático Medio (ASE) [10] en los datos de test que es un estimador que mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima. Su fórmula es la que se muestra en la figura 3.2

$$ASE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{3.2}$$

Capítulo 4

Variables

En este capitulo se definirán las variables extraídas de las diferentes fuentes de datos, se explicará el proceso de creación de nuevas variables y se hará un análisis descriptivo de las mismas.

4.1. Definición

Uno de los objetivos de este TFM es la creación de un cuadro de mando con el que poder visualizar toda la información, por lo que además de crear nuevas variables que nos ayuden a entender la problemática, también se harán transformaciones con la finalidad de poder mostrar los datos de una forma más clara en la aplicación.

La información que se extrae de la BTS está compuesta por 1361376 observaciones y las variables que se detallan en la siguiente tabla:

A las variables presentadas en la Tabla 4.1 se les aplican diferentes transformaciones como cambiar la distancia original (en millas) a Kilómetros, pasar el tiempo de vuelo al formato HH:MM, cambios de formato en las horas de salida y llegada y otras transformaciones. También se generan nuevas variables como puede ser 'esRetraso' que indica con 1 ó 0 si el vuelo ha tenido o no retraso respectivamente. Todo esto se hace en el script de carga de la aplicación. En el Código 4.1 se muestran algunas líneas que ejemplifican estas transformaciones.

Variable	**Tipo**	**Descripción**
YEAR	Nominal	Año del Vuelo
MONTH	Nominal	Mes del vuelo
DAY_OF_MONTH	Nominal	Día del mes del vuelo
AIRLINE	Nominal	Código de la aerolínea
TAIL_NUM	Nominal	Número del avión
FL_NUM	Nominal	Número de vuelo
ORIGIN_AIRPORT_ID	Nominal	Código del aeropuerto de origen
ORIGIN	Nominal	Origen
ORIGIN_CITY_NAME	Nominal	Ciudad de origen
ORIGIN_STATE_NM	Nominal	Estado de origen
DEST	Nominal	Destino
DEST_CITY_NAME	Nominal	Ciudad de destino
DEST_STATE_NM	Nominal	Estado de destino
CRS_DEP_TIME	Nominal	Hora de salida programada
DEP_TIME	Nominal	Hora de salida real
DEP_DELAY	Intervalo	Retraso en la salida
CRS_ARR_TIME	Nominal	Hora de llegada programada
ARR_TIME	Nominal	Hora de llegada real
ARR_DELAY	Intervalo	Retraso en la llegada
CANCELLED	Binaria	Cancelado
AIR_TIME	Intervalo	Tiempo de vuelo
DISTANCE	Intervalo	Distancia recorrida
CARRIER_DELAY	Intervalo	Tiempo de retraso por motivo del operador
WEATHER_DELAY	Intervalo	Tiempo de retraso por motivo meteorológico
NAS_DELAY	Intervalo	Tiempo de retraso por motivo del National Air System
SECURITY_DELAY	Intervalo	Tiempo de retraso por motivos de seguridad
LATE_AIRCRAFT_DELAY	Intervalo	Tiempo de retraso total en minutos

Tabla 4.1: Variables extraídas de la BTS

```
Num(SubField([DISTANCE],'.',1)/0.62137,'#.##0,00') as [Distancia KM
    ],
Time(TimeStamp#(If(len([CRS_DEP_TIME])=3, 0&[CRS_DEP_TIME],[
    CRS_DEP_TIME]),'hhmm'),'hh:mm') AS [Hora Salida],
Interval(SubField([AIR_TIME],'.',1)/24/60,'hh:mm') AS [Tiempo de
    Vuelo],
If([ARR_DELAY]>0, 1, 0) AS [esRetraso]
```

Código 4.1: Transformaciones en Qlik Sense.

Se realizan transformaciones con el objetivo de mejorar los modelos predictivos, como puede ser la agrupación de las variables de hora de salida o llegada, que se agrupan en 5 categorías como se muestra en el Código 4.2 en el lenguaje de programación de SAS.

De esta manera se reduce el número de categorías en esta variable (más de 60 en el momento inicial) con lo que también, por ejemplo, bajaremos el tiempo de ejecución en las redes neuronales.

```
DATA BASEANALISIS;SET BASEANALISIS;format Hora_Llegada_Agr $250.;
IF Hora_Llegada IN (7,8,9,10,11)
THEN Hora_Llegada_Agr='MANANA';
IF Hora_Llegada IN (12,13,14,15,16)
THEN Hora_Llegada_Agr='MEDIO DIA';
IF Hora_Llegada IN (17,18,19,20,21)
THEN Hora_Llegada_Agr='TARDE';
IF Hora_Llegada IN (22,23,0)
THEN Hora_Llegada_Agr='NOCHE';
IF Hora_Llegada IN (1,2,3,4,5,6)
THEN Hora_Llegada_Agr='MADRUGADA';
RUN;
```

Código 4.2: Transformaciones de la variable Hora Llegada en SAS.

De la web `http://ourairports.com/` se extraen las siguientes variables:

Variable	Tipo	Descripción
type	Nominal	Clasificación del aeropuerto
latitude_deg	Intervalo	Latitud del aeropuerto
longitude_deg	Intervalo	Longitud del aeropuerto
continent	Nominal	Continente del aeropuerto
iso_country	Nominal	País del aeropuerto
municipality	Nominal	Municipio del aeropuerto
iata_code	Nominal	Código del aeropuerto

Tabla 4.2: Variables referentes a aeropuertos

Las variables de la Tabla 4.2 servirán para clasificar los diferentes tipos de aeropuertos de destino y para poder representar en gráficos de mapa la situación de los aeropuertos.

Para esto se utiliza el Código 4.3 con el que se convierte la longitud y latitud a un GeoPoint de Qlik.

```
1 GeoMakePoint(latitude_deg, longitude_deg) as [Localizacion origen]
```

Código 4.3: Generación de Geo Point de Qlik.

La información meteorológica está compuesta por las variables que se muestran en la Tabla 4.3.

Variable	Tipo	Descripción
AWND	Intervalo	Intensidad del viento medio
PRCP	Intervalo	Precipitaciones
SNOW	Intervalo	Nieve
TAVG	Intervalo	Temperatura media
TMAX	Intervalo	Temperatura máxima
TMIN	Intervalo	Temperatura mínima
WSF2	Intervalo	Viento más rápido en 2 minutos
WSF5	Intervalo	Viento más rápido en 5 minutos

Tabla 4.3: Variables extraídas de los datos meteorológicos

Del proyecto de GitHub del usuario shivaas que se encuentra en la web `https://gist.github.com/shivaas/4758439` se extraen 2 variables, la festividad y la fecha, y se genera una nueva que marcará el tipo de día como se indicaba en la sección 2.1 Naturaleza de los datos.

Todas las variables presentadas anteriormente son las que se han extraído de las dife-

rentes fuentes de datos, pero no todas estas variables se utilizarán para realizar los modelos predictivos. Todas las variables que tienen una relación directa con la variable objetivo (que será el *Retraso_minutos*) y las variables que sean equivalentes, como por ejemplo, el código del aeropuerto de origen y el origen, no se utilizarán en las diferentes técnicas de minería de datos. Estas variables son las siguientes:

- ORIGIN_AIRPORT_ID
- EP_TIME
- DEP_DELAY
- ARR_TIME
- ARR_DELAY
- CARRIER_DELAY
- WEATHER_DELAY
- NAS_DELAY
- SECURITY_DELAY
- LATE_AIRCRAFT_DELAY

Variable objetivo

La variable que se utilizará como objetivo en la búsqueda del mejor modelo predictivo es *Retraso_minutos*. Esta variable se genera en el script de carga de la aplicación y se hace con el Código 4.4 donde restamos la hora de llegada real menos la hora de llegada programada. Por esto, la variable puede tomar valores negativos o positivos, indicando que el vuelo llega antes de tiempo cuando el resultado de la operación es negativo y que se retrasa si es positivo.

```
interval([Hora Llegada Real]-[Hora Llegada],'mm') as [Retraso
    minutos]
```

Código 4.4: Generación de la variable objetivo.

4.2. Análisis descriptivo

Para realizar el análisis descriptivo de las variables presentadas en la sección anterior, se utiliza la herramienta Qlik Sense.

Se comienza con una representación general de los datos. Esta representación se muestra en la Figura 4.1. En el gráfico de bloques de la izquierda se representan los aeropuertos de origen, este gráfico muestra que el aeropuerto de Atlanta es el que tiene más retrasos. Se representan los 25 aeropuertos de destino con más retrasos en el gráfico de bloques de la derecha y podemos ver como San Francisco es con diferencia el aeropuerto que tiene más retrasos.

En la parte inferior izquierda se muestra un histograma de la variable objetivo donde vemos que la mayoría de los vuelos tienen un retraso en minutos de $-35 <= x < 40$, es decir, la mayoría de los vuelos (1,2M) llegan entre 35 minutos antes de su hora y 40 después. El gráfico inferior derecha nos muestra un histograma enfrentando el tiempo de vuelo contra el retraso de los vuelos. Vemos que los vuelos más largos no son los que más se retrasan, esto podría ser debido a que tienen mucho más tiempo de vuelo para recuperar el retraso.

Se analiza la información por bloques, comenzaremos por la información de los destinos. Para estudiar la información por aeropuerto se crea la métrica Nº Retrasos / Nº Vuelos, esto nos dará el porcentaje de retrasos que se producen sobre el total de los vuelos que tiene cada aeropuerto. En base a esto podemos ver en la Figura 4.2 claramente como si sólo analizamos el número de retrasos todo los aeropuertos son de tipo *large_airport* pero si lo hacemos con la nueva métrica (Nº Retrasos / Nº Vuelos) se ve como los aeropuertos con mayor porcentaje de retrasos son de tipo *medium_airport* e incluso aparece alguno *small_airport*.

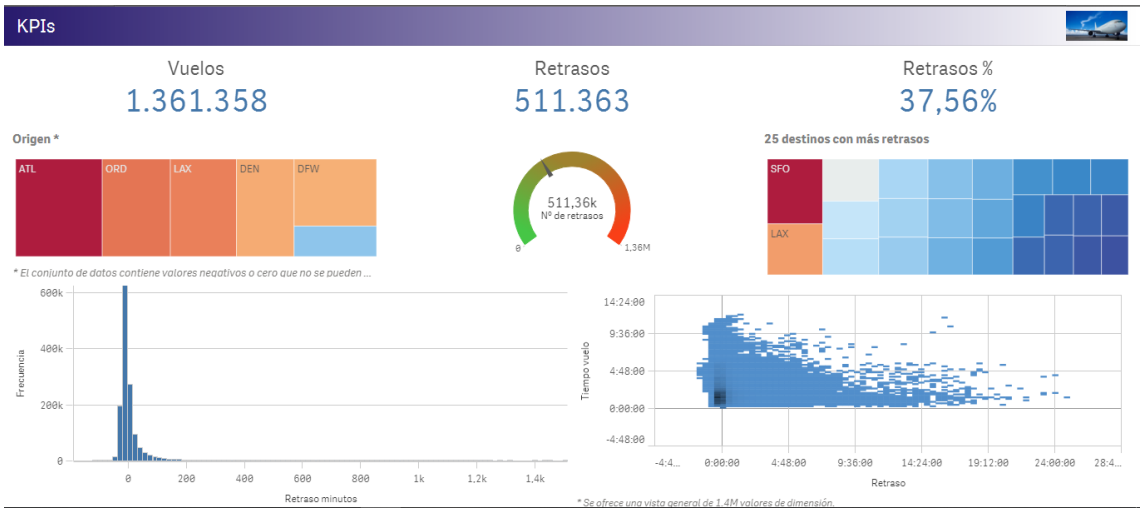


Figura 4.1: Hoja principal.

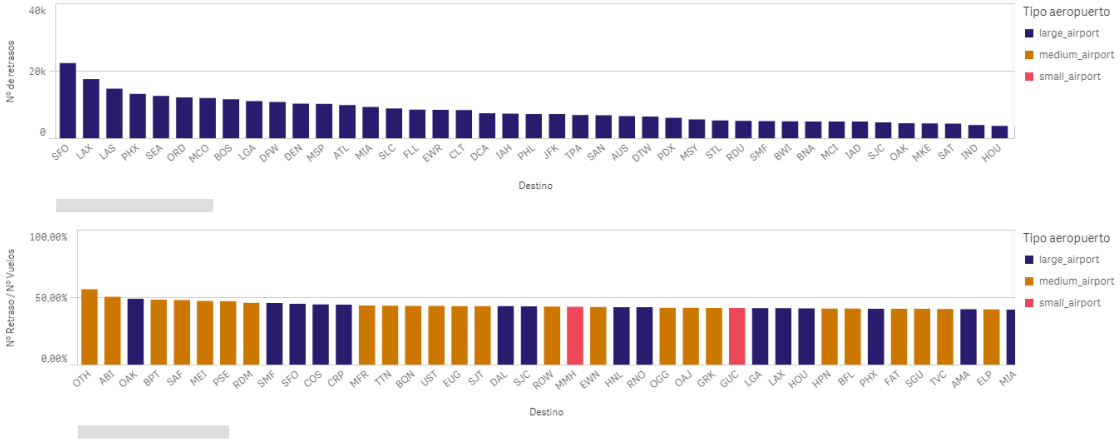


Figura 4.2: Comparativa destinos.

También se busca información según la zona en la que está el aeropuerto de destino. Como se muestra en la Figura 4.3 los retrasos se concentran en las ciudades del perímetro del país siendo las del centro las que menos retrasos tienen. El número de retrasos lo marca el tamaño de la burbuja y el color nos muestra el tipo del aeropuerto.

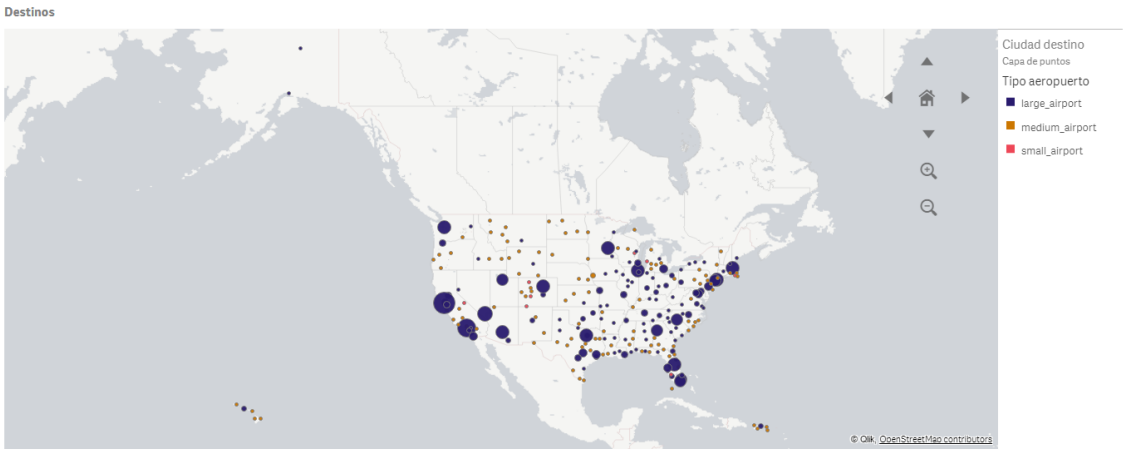


Figura 4.3: Mapa de destinos.

Otro bloque que se estudia es la parte temporal. Como vemos en la Figura 4.4, en la hoja principal de este bloque podemos ver los días con más retrasos en el gráfico superior coloreado por tipo de día y en la parte inferior los gráficos de bloques con la métrica Nº Retrasos / Nº Vuelos con el que ya vemos que el 3º trimestre es el que mayor porcentaje de retrasos se producen. El peor mes es Julio, seguido de Diciembre, Agosto y Junio. Los peores días de la semana son Jueves, Viernes y Lunes.

Figura 4.4: Hoja temporal.

Así podríamos analizar, por ejemplo, la probabilidad de coger un vuelo con retraso

un Sábado de Noviembre que es del 25 %, y por el contrario el día que más probabilidad tendríamos de coger un vuelo con retraso sería un Jueves de Julio con un 51 % de probabilidad. Este análisis se muestra en la imagen 4.5

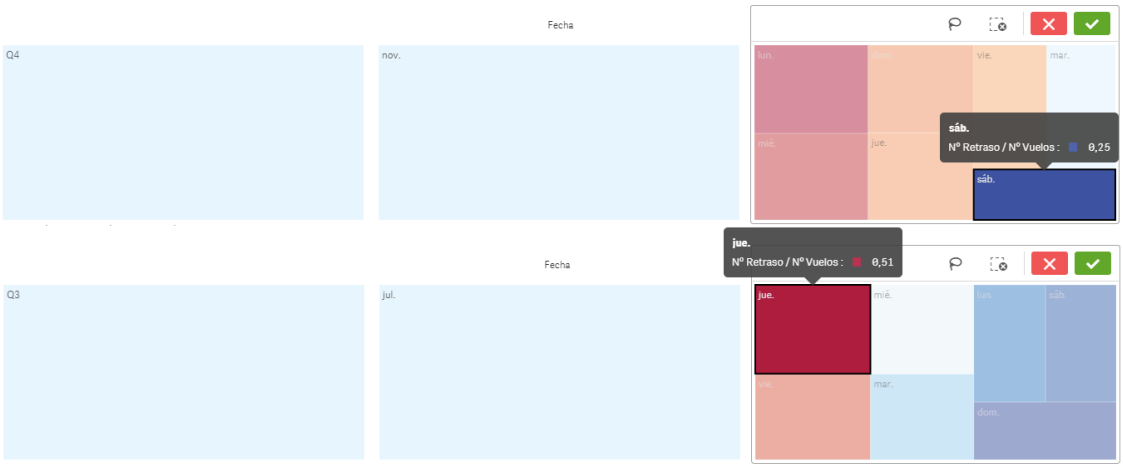


Figura 4.5: Análisis diario.

El último bloque de estudio es la meteorología. En este bloque cruzaremos el número de retrasos con las distintas condiciones meteorológicas para ver en forma de gráfico si hay dependencia entre estas variables. Se ve claramente en la Figura 4.6 como hay una fuerte dependencia del número de retrasos y las 3 principales condiciones meteorológicas, viento, lluvia y nieve. Se ve claramente como, por ejemplo, el día 23/03/2016 hay un máximo en los datos de nieve, y al día siguiente se dispara el dato de retrasos.

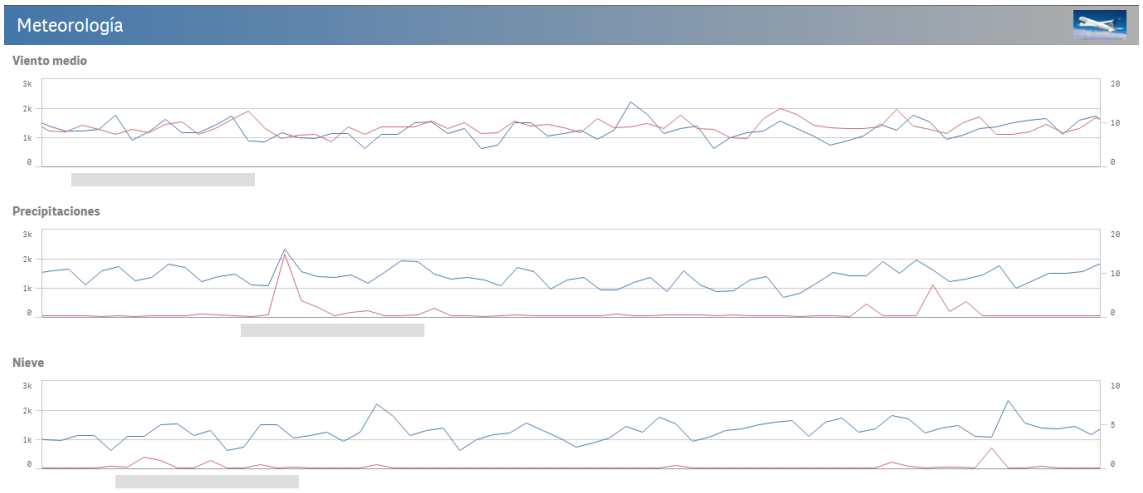


Figura 4.6: Gráficos Nº Retrasos y condiciones meteorológicas.

Lo que se obtiene al final es una aplicación con 3 bloques claramente diferenciados (Destino, Temporal y Meteorología) que nos sirve para estudiar el conjunto de datos en profundidad. Con la funcionalidad asociativa de Qlik podremos hacer un estudio más exhaustivo de las diferentes dimensiones establecidas en la aplicación, es decir, podríamos seleccionar un aeropuerto de origen y toda la aplicación mostraría sólo sus datos. De esta manera tenemos la opción de ir bajando en el nivel de detalle hasta llegar por ejemplo a días en concreto como hemos hecho con la parte meteorológica.

Con esta herramienta se consigue extraer la información que se ha obtenido de las diferentes fuentes de datos y como paso final, mediante el script de carga, se crea un archivo con únicamente las variables que se utilizarán para la búsqueda del mejor modelo predictivo.

Las variables de intervalo que se incluirán en los modelos se muestran en la Figura 4.7 con sus estadísticos principales.

Estadísticos descriptivos de la variable de intervalo

Variable	Etiqueta	Ausente	N	Mínimo	Máximo	Media	Desviación estándar	Asimetría	Curtosis
Distancia_KM		0	1361376	107.83	8019.38	1465.14	1032.99	1.2331	1.46
Nieve	Nieve	4170	1357206	0.00	30.30	0.03	0.59	31.5626	1311.87
Precipitaciones	Precipitaciones	0	1361376	0.00	83.80	0.75	4.49	9.7251	120.58
Retraso_minutos		18055	1343321	-107.00	1505.00	4.89	41.53	6.5696	92.50
Temperatura_m_nima	Temperatura mínima	0	1361376	-15.00	82.00	46.76	19.88	-0.3491	-0.98
Temperatura_m_xima	Temperatura máxima	0	1361376	3.00	104.00	65.21	22.27	-0.4938	-0.89
Temperatura_media	Temperatura media	196049	1165327	-1.00	88.00	53.38	20.72	-0.3807	-1.11
Tiempo_de_Vuelo_minutos		18055	1343321	10.00	723.00	122.58	77.40	1.3032	1.86
Viento_2min	Viento 2min	0	1361376	4.00	53.90	17.70	7.13	0.8484	1.33
Viento_5min	Viento 5min	5051	1356325	5.40	74.00	22.77	9.13	0.8009	1.24
Viento_medio	Viento medio	0	1361376	0.70	25.05	7.67	3.57	0.8032	1.26

Figura 4.7: Estadísticos descriptivos de las variables de intervalo.

Y las variables categóricas se muestran en la Figura 4.8 donde vemos el número de clases que tienen.

Estadísticos de sumarización de la variable de clase

Variable	Etiqueta	Tipo	Número de niveles	Ausente
Aerolinea		C	12	0
Cancelado		N	2	0
Ciudad_Destino		C	247	0
Cuatrimestre	Cuatrimestre	C	4	0
D_a_semana	Día semana	C	7	0
Festividad	Festividad	C	10	1193654
Hora_Llegada_Cat		C	5	32403
Hora_Salida_Cat		C	5	4297
Mes	Mes	C	12	0
Regi_n_origen	Región origen	C	6	0
Tipo_D_a	Tipo Día	C	6	0
Tipo_aeropuerto		C	3	0

Figura 4.8: Estadísticos descriptivos de las variables de clase.

Se tratarán las variables como *Ciudad_Destino* agrupando las categorías ya que, por ejemplo, esta variable tiene 247 clases y ciertos algoritmos de predicción como las redes neuronales no funcionan bien con un gran número de categorías. La variable *Festividad* se elimina ya que contiene el motivo de la festividad del día, por esto tiene tantos valores ausentes, en su lugar se utilizará *Tipo_dia* que nos aporta la misma información.

El coeficiente de correlación de Pearson es un índice que puede utilizarse para medir el grado de relación de dos variables siempre y cuando ambas sean cuantitativas. En la Figura 4.9 se muestra un gráfico de este coeficiente que nos da una idea de cómo afectan

las variables de entrada a la objetivo *Retraso_minutos*. Se observa que las variables que más influyen sobre el retraso que se comete en un vuelo de forma positiva [1] son las meteorológicas, es decir, cuanto más nieve, o más precipitaciones, o más viento, los vuelos se retrasarán más. Por el contrario, la variable *Distancia_KM* afecta negativamente a la variable objetivo, esto quiere decir que cuanto mayor sea el recorrido de la ruta aérea, menos retraso se producirá. La razón de esto es que en vuelos largos los retrasos que se producen antes de la salida del vuelo se puede recuperar. Para utilizar este nodo en SAS Miner (multi gráfico) se ha tenido que eliminar la variable *Ciudad_Destino* ya que no devuelve ningún resultado con variables con un gran número de categorías.

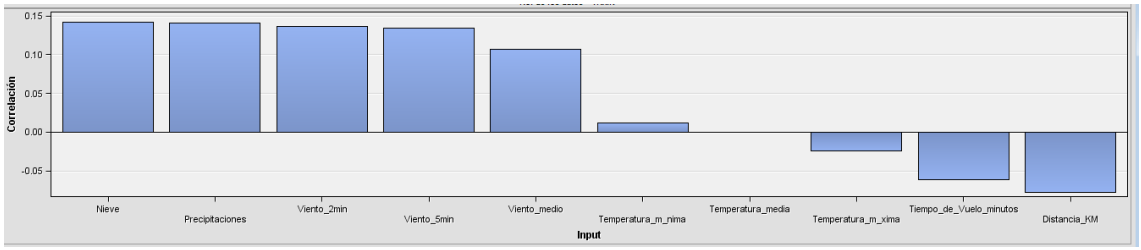


Figura 4.9: Gráfico de correlación de Pearson.

Combinando la utilización de las herramientas Qlik Sense y SAS Miner se realiza un análisis descriptivo con dos enfoques diferenciados. Qlik Sense nos da una visión más enfocada a negocio, donde podremos navegar por la información hasta encontrar patrones de comportamiento de los datos y SAS Miner nos muestra los datos con una clara visión estadística, dónde su finalidad sería realizar minería de datos para encontrar el mejor modelo predictivo.

[1] Hablar de que una variable afecta de forma positiva sobre la objetivo significa que aumenta, con lo cual habrá más retraso.

Capítulo 5

Modelado

Se muestran los diferentes modelos predictivos que se han generado y evaluado. En la Sección 3.2.1 se da una descripción teórica de las diferentes técnicas de modelado y, en este capítulo, se detallará de una forma práctica.

Para la búsqueda del mejor modelo se utilizan los programas estadísticos SAS Miner para la regresión lineal y SAS Base para los modelos de red neuronal, random forest, gradient boosting y ensamblado.

5.1. Training-Test

Para el estudio de los diferentes modelos predictivos se contemplaban dos técnicas de fragmentación de los datos, training-test o validación cruzada. Para este trabajo en concreto se decide utilizar una partición de los datos training-test [11] ya que el volumen de datos es grande para el hardware del que se dispone y los tiempos de ejecución con validación cruzada son muy superiores a training-test.

Este método consiste en dividir en dos conjuntos complementarios los datos de la muestra, realizar el análisis de un subconjunto (denominado datos de entrenamiento o training), y validar el análisis en el otro subconjunto (denominado datos de prueba o test), de forma que la función de aproximación sólo se ajusta con el conjunto de datos de entrenamiento y a partir de aquí calcula los valores de salida para el conjunto de datos

de prueba (valores que no ha analizado antes), de esta forma, se consigue una evaluación más realista del modelo.

En las técnicas de modelado que se expondrán a continuación se realiza una partición 80 %-20 % de forma aleatoria.

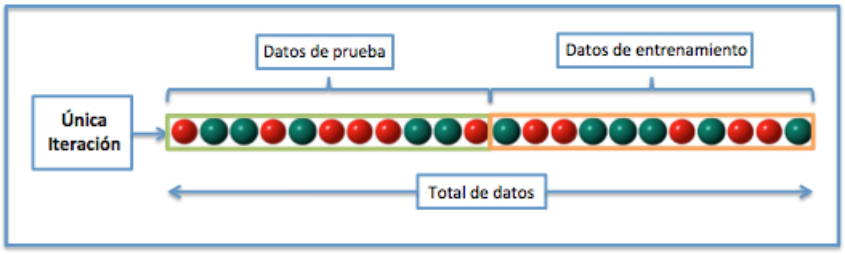


Figura 5.1: Ejemplo de división de un conjunto de datos en training-test.

5.2. Regresión lineal

La búsqueda del mejor modelo de regresión lineal se hará con el software estadístico SAS Miner. La razón principal por la que se utilizará este programa es que se probarán tres modelos de selección de variables:

- ***Stepwise***: En cada paso del algoritmo se evaluarán las posibles variables a eliminar y a introducir y se seleccionará aquella con mejor p-valor.

- ***Forward***: Este método va introduciendo una a una las variables que mayor mejora produzcan hasta que no haya ninguna variable que aporte información.

- ***Backward***: Parte del modelo que contiene todas las variables y va eliminando una a una las que menos influyan en el modelo hasta que todas las variables sean significativas.

Y tres criterios de parada *BIC*, *AIC* y *SBC*, lo que hace un total de 9 modelos, por lo que crear estos modelos en SAS Miner se hace de una forma sencilla y a nivel de representación de las salidas del modelo nos da más opciones que SAS Base.

Se utilizará el nodo de regresión lineal con los parámetros anteriormente descritos, por lo que serán 9 los nodos como se muestra en la Figura 5.2.

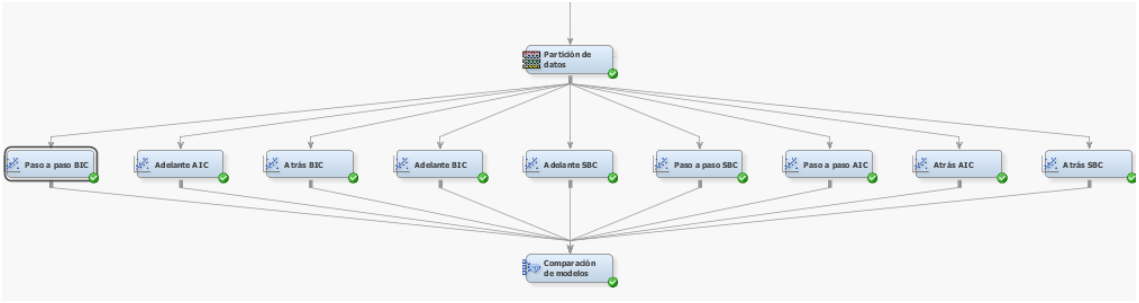


Figura 5.2: Regresión lineal en SAS Miner.

Con el nodo comparador de modelos evaluando el ASE en los datos test obtenemos los resultados de la Figura 5.4

Modelo seleccionado	Descripción del modelo	Variable target	Criterio de selección: Probar: error cuadrático medio
Y	Atrás BIC	Retraso_minutos	1650.891
	Atrás SBC	Retraso_minutos	1650.891
	Atrás AIC	Retraso_minutos	1650.905
	Adelante BIC	Retraso_minutos	1650.91
	Adelante SBC	Retraso_minutos	1652.727
	Adelante AIC	Retraso_minutos	1652.727
	Paso a paso SBC	Retraso_minutos	1663.974
	Paso a paso AIC	Retraso_minutos	1663.974
	Paso a paso BIC	Retraso_minutos	1669.302

Figura 5.3: ASE de los modelos de regresión lineal.

Los errores de los 9 modelos son similares, siendo el de los primeros dos que se muestran en la Figura 5.4 iguales y más bajos. Seleccionamos como mejor modelo la regresión lineal con el **modelo de selección de variables *backward*** con **criterio de parada *BIC*** que tiene un ASE de **1650,89**.

Para el estudio de esta técnica se emplean interacciones entre las variables. Con esto, el modelo se produce en la 17ª iteración del algoritmo por lo que las variables que producen la regresión lineal son las que se muestran en la Tabla 5.1. En la primera columna se muestran las variables seleccionadas en el modelo. De cara a mejorar el modelo, se generan todas las interacciones de grado 2 de dichas variables y se seleccionan. Para este modelo de regresión

lineal son las 36 que se muestran en la segunda columna.

El modelo de regresión lineal es el compuesto por todas estas variables más todas las interacciones.

Se utiliza la variable *Aerolínea* como ejemplo para explicar los estimadores del modelo. Como se observa en la Figura 5.4, el estimador indica que un vuelo operado por las aerolíneas *Alaska Airlines Inc.* o *Hawaiian Airlines Inc.* influye de forma negativa en la variable objetivo que es el retraso en minutos, es decir, habrá menos retraso en estos vuelos. Por el contrario, si el vuelo lo cubre *JetBlue Airways* o *Spirit Air Lines* habrá más retraso ya que estas aerolíneas tienen un estimador positivo.

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Valor t	Pr > |t|
Intercept		1	-21.7729	5.0993	-4.27	<.0001
Aerolinea	Alaska Airlines Inc.: AS	1	-5.8686	1.5320	-3.83	0.0001
Aerolinea	American Airlines Inc.:	1	0.0473	0.9676	0.05	0.9610
Aerolinea	Delta Air Lines Inc.: DL	1	-2.8595	0.9618	-2.97	0.0029
Aerolinea	ExpressJet Airlines Inc.	1	-0.7855	1.1275	-0.70	0.4860
Aerolinea	Frontier Airlines Inc.:	1	3.3444	1.1506	2.91	0.0037
Aerolinea	Hawaiian Airlines Inc.:	1	-4.5520	3.0020	-1.52	0.1294
Aerolinea	JetBlue Airways: B6	1	4.6661	1.1168	4.18	<.0001
Aerolinea	SkyWest Airlines Inc.: 0	1	1.0780	1.0516	1.03	0.3053
Aerolinea	Southwest Airlines Co.:	1	1.2431	0.9798	1.27	0.2045
Aerolinea	Spirit Air Lines: NK	1	4.1170	1.1503	3.58	0.0003
Aerolinea	United Air Lines Inc.: U	1	-1.3309	0.9666	-1.38	0.1686
Aerolinea	Virgin America: VX	0	0	.	.	.

Figura 5.4: Estimadores de máxima verosimilitud de la variable *Aerolínea*.

5.3. Redes neuronales

Para conseguir la red neuronal que menos error cometa en los datos de test se ejecutan diferentes bucles con los que iremos variando los parámetros del procedimiento *neural* [1] de SAS, de este modo tendremos en una tabla los diferentes modelos de red neuronal con su ASE.

Se utilizará en los modelos de red neuronal *early stopping* que nos ayudará a no sobreajustar la red.

Los parámetros que se modificarán en bucle del *proc neural* son:

[1]Documentación *neural*: `http://documentation.sas.com/?docsetId=inmsref&docsetTarget=p0o8wrmmp8zkisn1riwf74uvagq9.htm&docsetVersion=2.81&locale=en`.

Variables
Aerolinea
Ciudad_Destino
Dia_semana
Distancia_KM
Festividad
Hora_Llegada_Cat
Hora_Salida_Cat
Nieve
Precipitaciones
Region_origen
Temperatura_maxima
Temperatura_media
Tiempo_de_Vuelo_minutos
Tipo_Dia
Viento_2min
Viento_medio

Interacciones de dos variables
Distancia_KM*Distancia_KM
Distancia_KM*Nieve
Distancia_KM*Precipitaciones
Distancia_KM*Temperatura_minima
Distancia_KM*Temperatura_maxima
Distancia_KM*Temperatura_media
Distancia_KM*Tiempo_de_Vuelo_minutos
Distancia_KM*Viento_5min
Distancia_KM*Viento_medio
Nieve*Nieve
Nieve*Precipitaciones
Nieve*Temperatura_minima
Nieve*Tiempo_de_Vuelo_minutos
Precipitaciones*Precipitaciones
Precipitaciones*Temperatura_minima
Precipitaciones*Temperatura_media
Precipitaciones*Tiempo_de_Vuelo_minutos
Precipitaciones*Viento_2min
Temperatura_minima*Temperatura_maxima
Temperatura_minima*Temperatura_media
Temperatura_minima*Tiempo_de_Vuelo_minutos
Temperatura_minima*Viento_2min
Temperatura_minima*Viento_5min
Temperatura_minima*Viento_medio
Temperatura_maxima*Temperatura_media
Temperatura_maxima*Tiempo_de_Vuelo_minutos
Temperatura_maxima*Viento_2min
Temperatura_maxima*Viento_medio
Temperatura_media*Tiempo_de_Vuelo_minutos
Temperatura_media*Viento_2min
Temperatura_media*Viento_5min
Tiempo_de_Vuelo_minutos*Tiempo_de_Vuelo_minutos
Viento_2min*Viento_2min
Viento_2min*Viento_5min
Viento_2min*Viento_medio

Tabla 5.1: Variables e interacciones seleccionadas en el modelo de regresión lineal.

- **Algoritmo de optimización:** Son 6 algoritmos que iremos combinando con los demás parámetros, Levmar, QuaNew, ConGra, DBLDog, BProp y TruReg.

- **Función de activación:** La función activación [12] calcula el estado de actividad de una neurona; transformando la entrada global en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (–1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o –1) o activa (1). A continuación se muestran los parámetros que se van a probar en el *proc neural* con la función de activación que aplican a la red neuronal:

  - Lin: Lineal t
  - Arc: Arcotangente $arctan(t) \cdot \frac{2}{\pi}$
  - Sin: Seno $sin(t)$
  - Sof: Función exponencial normalizada $\frac{e^t}{\sum exponentials}$
  - Gau: Gaussiana e^{-t^2}
  - Tanh: Tangente $tanh(t) = 1 - \frac{2}{(1+e^{(2t)})}$
  - Log: Logística $\frac{1}{1+e^{-t}}$

- **Número de nodos ocultos:** Las diferentes redes que se irán probando tendrán una capa oculta e iremos variando el número de nodos. Este parámetro irá creciendo desde 2 hasta 20 de 2 en 2.

Las redes neuronales consumen muchos recursos computacionales por lo que en lugar de iterar todos los parámetros entre si, se hará de una forma escalonada, es decir, primero se probará con el número de nodos ocultos para un algoritmo de optimización Levmar y una función de activación Tanh.

De esta manera se obtienen 10 modelos. En la Tabla 5.2 se muestra el ASE cometido por cada uno de ellos cambiando en número de nodos y manteniendo como algoritmo de optimización Levmar y como función de activación Tanh. El cálculo del error se aplica sobre

Número de nodos	**ASE**
2	1578,19
4	1592,91
6	1574,95
8	1584,08
10	1585,64
12	1576,97
14	1574,90
16	**1568,55**
18	1585,91
20	1581,11

Tabla 5.2: Modelos de red neuronal cambiando el número de nodos.

Algoritmo de optimización	**ASE**
Levmar	**1568,55**
QuaNew	1609,86
ConGra	1625,06
DBLDog	1620,56
BProp	1712,49
TruReg	1588,56

Tabla 5.3: Resultados variando el algoritmo de optimización.

el conjunto de datos test de la función de salida score que se ejecuta en el procedimiento *neural*.

En base a los resultados obtenidos se escoge como número de nodos ocultos 16 que comete un error de 1568,55. Con estos nodos pasaremos a probar los diferentes algoritmos de optimización con la función de activación Tanh, con lo que obtenemos otros 6 modelos cuyos resultados se muestran en la Tabla 5.3.

Por último, se probará con las 7 funciones de activación descritas anteriormente, otras 7 iteraciones lo que harán un total de 21 redes neuronales. Los resultados de estas últimas iteraciones se muestra en la Tabla 5.4

Con todo esto, el mejor modelo de red neuronal es la que tiene como número de **nodos ocultos 16**, el **algoritmo de optimización Levmar** y como **función de activación Arc** con un **ASE de 1563,95**. Este modelo será el que compararemos con la regresión, random forest y gradient boosting.

Funciones de activación	ASE
Lin	1590,88
Arc	**1563,95**
Sin	1568,94
Sof	1583,27
Gau	1598,85
Tanh	1568,55
Log	1585,53

Tabla 5.4: Resultados para las diferentes funciones de activación.

Cabe decir que no se han ejecutado más modelos de red neuronal por las limitaciones de hardware para la realización de este TFM ya que al tener en el conjunto de datos variables con muchas categorías (se tuvieron que agrupar para la ejecución de estos modelos) que se mostraron en la Sección 4.1 los tiempos de ejecución eran muy malos teniendo que esperar más de 96 horas para finalizar la ejecución del procedimiento *neural* en algunos casos.

5.4. Random forest

Random forest trata de incorporar dos fuentes de variabilidad (remuestreo de observaciones y de variables) para ganar en capacidad de generalización, y reducir el sobreajuste conservando a la vez la facultad de ajustar bien relaciones particulares en los datos (interacciones, no linealidad, cortes, problemas de extrapolación, etc.).

El mejor modelo de random forest lo buscaremos utilizando el procedimiento de SAS *hpforest* [2]. Se procederá de la misma forma que con el procedimiento *neural*, es decir, se crearan macros con las que iterar los diferentes parámetros que el procedimiento admite para conseguir el modelo con el menor ASE. Los parámetros que se utilizarán para este modelo son los siguientes:

- **Número de árboles**: Se probará desde 10 hasta 210 árboles aumentándolos de 50 en 50. El parámetro que marcará el número de árboles que se utilizan es *maxtrees*.
- **Profundidad máxima del árbol**: Se probarán en este parámetro los valores de 15

[2]Documentación *hpforest*: `http://go.documentation.sas.com/?docsetId=emhpprcref&docsetVersion=14.2&docsetTarget=emhpprcref_hpforest_details.htm&locale=en`.

a 25 aumentando en intervalos de 5. El parámetro que indica la profundidad máxima es *maxdepth*.

- **Número de variables**: Es el parámetro *vars_to_try* del procedimiento *hpforest*, variará entre 2 y 18 de 2 en 2 sin llegar a utilizar todas ya que en lugar de random forest estaríamos haciendo bagging.

- **Tamaño mínimo de hoja**: Esto indica el número mínimo de observaciones que tiene que tener un nodo para ser una hoja. Los valores asignados al parámetro *leafsize* son de 100 a 2100 aumentando en cada iteración 500.

- **Porcentaje de datos en el remuestreo**: El parámetro que representa este porcentaje es *trainfraction*. El valor de este parámetro siempre será 0,5.

- **p-valor**: A este parámetro se le ha asignado el valor 0,1 que se corresponde con el parámetro *alpha* de construcción de árboles.

Los tiempos de ejecución de los modelos de random forest son asumibles por lo que se itera "todo con todo". Esto produce más de 750 modelos de random forest. En la figura 5.5 se muestra el resultado de los 25 mejores modelos de random forest. La columna Variables se corresponde con el parámetro *vars_to_try*, maxtrees con el parámetro *mactrees*, porcenbag es el parámetro *trainfraction*, tamhoja se corresponde a *leafsize*, maxdepth a la profundidad máxima del árbol y pvalor es el parámetro *alpha*. El error ASE se muestra en la columna que tiene como nombre media. Nos quedaremos con el mejor modelo para utilizarlo en ensamblado.

	media	Variables	maxtrees	porcenbag	tamhoja	maxdepth	pvalor
1	1724.4463664	2	110	0.5	2100	15	0.1
2	1724.4482994	2	160	0.5	2100	15	0.1
3	1724.7095416	2	110	0.5	2100	25	0.1
4	1725.0150153	2	210	0.5	2100	15	0.1
5	1725.1061733	2	110	0.5	2100	25	0.1
6	1725.1736819	2	210	0.5	2100	25	0.1
7	1725.2861934	2	110	0.5	2100	20	0.1
8	1725.4190273	2	210	0.5	2100	20	0.1
9	1725.6489973	2	210	0.5	2100	15	0.1
10	1725.6619472	2	210	0.5	2100	25	0.1
11	1726.0006456	2	60	0.5	2100	15	0.1
12	1726.0351895	2	160	0.5	2100	25	0.1
13	1726.1952653	2	210	0.5	2100	20	0.1
14	1726.2698707	2	60	0.5	2100	15	0.1
15	1726.3110848	2	160	0.5	2100	20	0.1
16	1726.4255257	2	110	0.5	2100	15	0.1
17	1726.464872	2	160	0.5	2100	20	0.1
18	1726.8147788	2	110	0.5	2100	20	0.1
19	1726.8489628	2	10	0.5	2100	25	0.1
20	1726.8560211	2	160	0.5	2100	25	0.1
21	1727.3001556	2	160	0.5	2100	15	0.1
22	1727.5211068	2	10	0.5	2100	20	0.1
23	1728.0298186	2	10	0.5	2100	15	0.1
24	1728.1084721	2	60	0.5	2100	25	0.1
25	1728.5778675	2	210	0.5	1600	25	0.1

Figura 5.5: 25 mejores modelos de random forest.

Estos 25 primeros modelos cometen un error muy similar pero nos quedaremos con el primero de la Figura 5.5 que tiene los siguientes parámetros:

- **Número de árboles**: *maxtrees*=110
- **Profundidad máxima del árbol**: *maxdepth*=15
- **Número de variables**: *vars_to_try*=2
- **Tamaño mínimo de hoja**: *leafsize*=2100
- **Porcentaje de datos en el remuestreo**: *trainfraction*=0,5
- **p-valor**: *alpha*=0,1

Con este modelo se comete un ASE de **1724,44**.

5.5. Gradient boosting

Esta técnica de aprendizaje automático consiste en repetir la construcción de árboles de regresión/clasificación, modificando ligeramente las predicciones iniciales cada vez, intentando minimizar los residuos en la dirección de decrecimiento.

Se ejecutará el procedimiento de SAS Base *treeboost* [3] variando sus parámetros de entrada para conseguir el modelo de gradient boosting con menos ASE.

Los parámetros del procedimiento *treeboost* que se irán cambiando en cada iteración son los siguientes:

- **Iteraciones**: Esta opción especifica el número de términos en boosting. Para variables objetivo de intervalo y binarias, el número de iteraciones es igual a la cantidad de árboles. El valor de este parámetro debe ser un número entero entre 1 y 1000 y por defecto está a 50 para variables objetivo de intervalo y binarias. El parámetro *iterations* se moverá en este caso entre 10 y 110 en intervalos de 50, por lo que tendremos 10, 60 y 110 iteraciones en cada ejecución.

- **Número de subconjuntos**: Esta opción especifica el número máximo de subconjuntos que puede producir una regla de división. Por ejemplo, si establece un número igual a 2, solo se producirán divisiones binarias en cada nivel. Si establece el número igual a 3, entonces son posibles las divisiones binarias o ternarias en cada nivel. El parámetro *maxbranch* variará entre 2 y 6 con incrementos de 2, es decir, se probará con 2, 4 y 6 números máximos de subconjuntos.

- **Profundidad máxima del árbol**: Esta opción determina la profundidad máxima de un árbol de decisión. La profundidad de un árbol es el número de reglas de división que son necesarias para llegar la hoja más "lejana". El nodo raíz tiene una profundidad de cero, mientras que sus hijos inmediatos tienen una profundidad de uno, y así sucesivamente. El procedimiento *treeboost* continuará buscando nuevas reglas de división siempre que la profundidad del nodo actual sea menor que el

[3] Documentación *treeboost*: `http://support.sas.com/documentation/solutions/emtmsas/93/emprcref.pdf`.

número. El valor predeterminado para el número es 6 y la opción MAX establece este valor en 50. Para buscar el mejor modelo de gradient boosting se variará el parámetro *maxdepth* entre 15 y 25 con incrementos de 5 obteniendo así 3 iteraciones.

- **Tamaño de hoja mínimo**: Especifica el número mínimo de observaciones que es necesario para formar una nueva rama. Este argumento especifica un número exacto de observaciones. Variaremos el valor del *leafsize* desde 100 hasta 1100 con incrementos de 500, por lo que serán 3 iteraciones.

- **Parámetro de Regularización**: Esta opción especifica cuánto, en porcentaje, se reduce la predicción de cada árbol. El valor del número debe estar entre 0 y 1 y el valor predeterminado es 0.2. Este parámetro *shrinkage* variará en nuestro caso entre los valores 0.01, 0.05, 0.09.

Al igual que pasaba con la técnica de random forest, gradient boosting tiene unos tiempos de proceso más que aceptables por lo que se va a iterar todo con todo, esto quiere decir, que se producirán más de 200 modelos de gradient boosting y según el ASE que cometan se elegirá el mejor modelo.

En la Figura 5.6 se muestran los 25 mejores modelos de esta técnica. Donde las columnas *maxbranch*, tamhoja se corresponde con *leafsize*, *shrink*, *iterations* y *maxdepth* se corresponden con los parámetros descritos anteriormente y la columna media se corresponde al ASE de cada modelo.

	maxbranch	tamhoja	shrink	iterations	maxdepth	media
1	2	100	0.09	60	25	1133.380255
2	4	100	0.09	60	25	1135.9299204
3	4	100	0.09	60	15	1137.2758476
4	2	100	0.09	60	20	1161.5031586
5	2	100	0.09	60	20	1161.5031586
6	2	100	0.09	110	15	1161.9669685
7	2	100	0.09	110	25	1161.9669685
8	4	100	0.09	110	15	1161.9669685
9	4	100	0.09	110	25	1161.9669685
10	2	100	0.05	60	25	1209.5517382
11	2	100	0.05	60	25	1209.5517382
12	2	100	0.05	60	25	1209.5517382
13	4	100	0.05	60	20	1213.2989103
14	4	100	0.05	60	20	1213.2989103
15	4	100	0.05	60	25	1213.5174837
16	4	100	0.05	60	25	1213.5174837
17	4	100	0.05	60	25	1213.5174837
18	2	100	0.05	110	15	1213.7969402
19	2	100	0.05	110	25	1213.7969402
20	4	100	0.05	110	15	1213.7969402
21	4	100	0.05	110	25	1213.7969402
22	4	100	0.05	60	15	1213.9452691
23	4	100	0.05	60	15	1213.9452691
24	4	100	0.05	60	15	1213.9452691
25	2	100	0.09	60	15	1221.3249646

Figura 5.6: 25 mejores modelos de gradient boosting.

El mejor modelo es el que contiene los siguientes parámetros:

- **Iteraciones**: *iterations*=60
- **Número de subconjuntos**: *maxbranch*=2
- **Profundidad máxima del árbol**: *maxdepth*=25
- **Tamaño de hoja mínimo**: *leafsize*=100
- **Parámetro de Regularización**: *shrinkage*=0.09

Que produce un ASE de **1133,38**.

5.6. Ensamblado

Para finalizar el estudio de las diferentes técnicas de minería de datos se propone realizar un modelo de ensamblado utilizando los mejores modelos de regresión lineal, red

neuronal, random forest y gradient boosting.

Para realizar el ensamblado se utilizará el método Stacking [9] con SAS Base. Se utiliza en general este término para cualquier tipo de combinación de modelos. Dadas las predicciones y_1, y_2, y_3 obtenidas por diferentes algoritmos, se combinan sus resultados. Existen tres opciones básicas:

1. Averaging (promediado): se calcula el promedio de las predicciones. Se puede utilizar también promedio ponderado, por ejemplo 0.80*predigbm+0.20*predirandomforest

2. Voto (para clasificación): se predice el resultado con mayoría entre las predicciones: $y_1 = 0, y_2 = 0, y_3 = 1 \rightarrow prediccion = 1$

3. Combinación a partir de otro algoritmo (esto es estrictamente stacking). Por ejemplo, se introducen en una regresión o árbol y_1, y_2, y_3 como variables independientes. En regresión equivaldría a un promediado de modelos con pesos diferentes.

En este caso concreto se utilizará la primera opción, averaging. Para ello se construye un conjunto de datos que contiene las predicciones de los cuatro modelos y en base a estos resultados se combinan las predicciones generando 11 nuevos modelos que se muestran a continuación.

Partiendo de que y_{reg} es la predicción de la regresión lineal, y_{nn} la predicción de la red neuronal, y_{rf} la predicción del random forest y y_{gb} las predicciones del gradient boosting:

- Regresión lineal con red neuronal: $y_{ens1} = (y_{reg} + y_{nn})/2$
- Regresión lineal con random forest: $y_{ens2} = (y_{reg} + y_{rf})/2$
- Regresión lineal con gradient boosting: $y_{ens3} = (y_{reg} + y_{gb})/2$
- Red neuronal con random forest: $y_{ens4} = (y_{nn} + y_{rf})/2$
- Red neuronal con gradient boosting: $y_{ens5} = (y_{nn} + y_{gb})/2$
- Random forest con gradient boosting: $y_{ens6} = (y_{rf} + y_{gb})/2$
- Regresión lineal, red neuronal y random forest: $y_{ens7} = (y_{reg} + y_{nn} + y_{rf})/3$

- Regresión lineal, red neuronal y gradient boosting: $y_{ens8} = (y_{reg} + y_{nn} + y_{gb})/3$
- Regresión lineal, random forest y gradient boosting: $y_{ens9} = (y_{reg} + y_{rf} + y_{gb})/3$
- Red neuronal, random forest y gradient boosting: $y_{ens10} = (y_{nn} + y_{rf} + y_{gb})/3$
- Regresión lineal, red neuronal, random forest y gradient boosting: $y_{ens11} = (y_{reg} + y_{nn} + y_{rf} + y_{gb})/4$

Con todas estas combinaciones se obtienen 11 modelos diferentes en los que tendremos la predicción media de cada observación. Calculando la media de la predicción de todas las observaciones obtendremos el ASE de cada modelo de ensamblado.

En la Figura 5.7 se muestran los modelos de ensamblado ordenado por su ASE.

	ASE	Modelo
1	1361.2567675	Regresion lineal y red neuronal
2	1377.0460084	Regresion lineal, red neuronal y gradient boosting
3	1387.3934966	Regresion lineal, red neuronal y random forest
4	1395.0718961	Regresion lineal, red neuronal, random forest y gradient boosting.
5	1497.5688511	Regresion lineal y gradient boosting
6	1519.362128	Regresion lineal random forest
7	1529.6382756	Regresion lineal, random forest y gradient boosting
8	1597.8124519	Red neuronal y gradient boosting
9	1602.9487636	Red neuronal, random forest y gradient boosting
10	1622.9042887	Red neuronal y random forest
11	1681.2851369	Random forest y gradient boosting.

Figura 5.7: ASE de los modelos de ensamblado.

El modelo de ensamblado que menor ASE tiene es el que combina la regresión lineal con la red neuronal con error de **1361,25**.

5.7. Selección del mejor modelo

En las secciones anteriores se han presentado las diferentes técnicas de minería de datos que se han utilizado en este TFM, con su explicación teórica, sus ventajas y desventajas, el estudio práctico de las técnicas con la variación de sus principales parámetros para conseguir el modelo que menos ASE produzca.

En la Tabla 5.5 podemos ver los 5 mejores modelos, uno por cada técnica presentada.

Técnica	Parámetros	ASE
Regresión Lineal	- Selección de variables: Backward - Criterio de parada: BIC	1650,89
Redes Neuronales	- Nodos ocultos: 16 - Algoritmo de optimización: Levmar - Función de activación: Arc	1563,95
Random Forest	- Número de árboles: 110 - Profundidad máxima del árbol: 15 - Número de variables: 2 - Tamaño mínimo de hoja: 2100 - Porcentaje de datos en el remuestreo: 0,5	1724,44
Gradient Boosting	**- Iteraciones: 60** **- Número de subconjuntos: 2** **- Profundidad máxima del árbol: 25** **- Tamaño de hoja mínimo: 100**	**1133,38**
Ensamblado	- Regresión lineal - Red neuronal	1361,25

Tabla 5.5: Mejores modelos de cada técnica de minería de datos.

Claramente podemos concluir que el mejor modelo se produce con la técnica gradient boosting presentada en la Sección 5.5 con un ASE en el conjunto de datos test de **1133,38**.

El coeficiente de determinación [13], denominado R^2, es un estadístico que determina la calidad del modelo para replicar los resultados, y la proporción de variación de los resultados que puede explicarse por el modelo.

Teniendo el ASE del modelo, para calcular el R^2 se utilizará la Fórmula 5.1 que devuelve una estimación de este estadístico.

$$R^2 \approx 1 - \frac{ASE}{\sigma^2} \tag{5.1}$$

Siendo la varianza de la variable objetivo de los datos de test 1692,50 se calcula el R^2 que da un resultado de **0,34** con lo que se puede concluir que el mejor modelo de las técnicas estudiadas en este trabajo explica aproximadamente un 34 % de la variable objetivo.

Las variables más importantes de este modelo son las que se muestran en la Figura 5.8. Este gráfico se genera al reproducir el mejor modelo de gradient boosting de SAS Base en

SAS Miner exportando todos los parámetros. Así podemos observar que para esta técnica de minería de datos con los parámetros anteriormente descritos las cinco variables que más importancia tienen en el modelo son *Mes*, *Precipitaciones*, *Hora_Llegada_Cat*, *Aerolinea* y *Hora_Salida_Cat*.

Lo que confirma lo que se describía en la Sección 4.2 sobre las variables meteorológicas y temporales tienen mucha influencia sobre la variable objetivo.

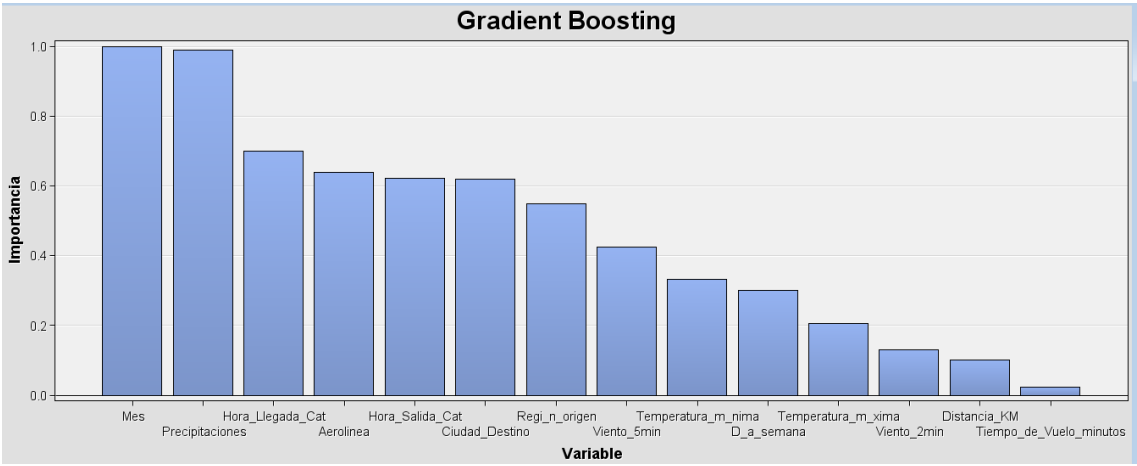


Figura 5.8: Variables más importantes del mejor modelo de gradient boosting.

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se hará una reflexión sobre el trabajo realizado y las posibles líneas de trabajo futuras.

En la Sección 6.1 se explican las conclusiones del proyecto una vez finalizado y en la Sección 6.2 se estudiarán las líneas futuras de trabajo sobre la aplicación.

6.1. Conclusiones

El objetivo de este trabajo es el desarrollo de una aplicación en Qlik Sense que nos muestre toda la información recopilada de las diferentes fuentes de datos de las que se ha hablado en la Sección 2.1 y la búsqueda del mejor modelo predictivo empleando diferentes técnicas de minería de datos.

Se han cumplido los objetivos que presentábamos en la Sección 3.1:

- Se consigue la realización de una aplicación en Qlik Sense intuitiva y sencilla dónde poder explotar toda la información recopilada.

- Se ha conseguido elaborar un modelo de datos asociativo con el que podemos relacionar todos los datos extraídos de las diferentes fuentes de datos y que nos ayudará a navegar por los datos de una forma fácil.

- Se puede explotar la información relativa a las diferentes rutas aéreas desde diversos puntos de vista, los cuales ayudan a entender las causas de los retrasos de los vuelos.

- Se han elaborado diversas métricas que ayudan a entender el cómo se produce un retraso, no conformándonos con una única métrica sobre el retraso en minutos de las rutas aéreas.

- La creación de diversos gráficos en la herramienta de Qlik Sense nos muestra toda la información desde diversos puntos de vista y con dimensiones y métricas cruzadas, lo que ofrece infinidad de oportunidades de encontrar información dentro de los datos.

- Estudiar las variables con una finalidad más orientada al modelado y búsqueda de los diferentes modelos predictivos que se aplicaron en este TFM.

- Se profundizó en la búsqueda del mejor modelo predictivo empleando diversas técnicas de minería de datos, consiguiendo un resultado comparable para la evaluación de los diferentes modelos y escogiendo en modelo con menor ASE. Este modelo es el presentado en la Sección 5.7 generado con la técnica gradient boosting. El valor de R^2 indica el porcentaje explicativo del modelo sobre la variable objetivo, en este caso es de un 34 %. Las diez variables que más influencia tienen en el modelo son *Mes, Precipitaciones, Hora llegada, Aerolínea, Hora Salida, Destino, Origen, Viento, Temperatura mínima y Día de la semana*. Esto nos confirma lo comentado en la Sección 4.2, las variables referentes a la climatología y el tiempo tienen una gran importancia sobre la variable objetivo.

- Se consigue crear diferentes macros en SAS Base para la generación de los modelos predictivos, tanto para la parametrización de los procedimientos propios de SAS, como para la automatización de la búsqueda de los mejores modelos utilizando bucles anidados que facilitaban iterar los diferentes parámetros. También se consiguieron adaptar diferentes macros construidas por el profesor Javier Portela García-Miguel para que fuesen funcionales con training-test y con una variable objetivo de intervalo.

Relativo a los conocimientos adquiridos sobre el software empleado, se profundizó en los tres programas empleados en la realización de este TFM que son Qlik Sense, SAS Base y SAS Miner y se adquirieron mejores habilidades de programación en lenguaje SAS.

Se adquirieron conocimientos relativos al desarrollo de un proyecto de minería de datos, además de profundizar en la metodología SEMMA que se describe en la Sección 3.2

Con todo esto la experiencia personal en este trabajo ha sido satisfactoria, ya no sólo por haber logrado los objetivos marcados antes de empezarlo, también por todos los conocimientos adquiridos que me ayudarán en un futuro.

6.2. Líneas futuras

Como la mayoría de trabajos de este carácter, existe la posibilidad de que cambien las variables que influyen sobre el tema. Podría darse el caso de que nuevas variables influyan en la determinación de el retraso de un vuelo comercial o que las que ahora mismo son importantes en esta predicción pierdan su importancia. Por todo esto, la actualización de la información, la revisión de los modelos predictivos y la adecuación del cuadro de mando de Qlik Sense a la realidad es esencial en un trabajo como este. Para que el trabajo aporte cada día más valor, es indispensable que sea capaz de adaptarse a todos estos cambios.

Como líneas de trabajo futuro, se destacan las siguientes características:

- Una gran limitación para este trabajo ha sido la inherente al hardware del que se disponía para la realización de los modelos predictivos. Esta limitación ha provocado que la ejecución de las diferentes técnicas de minería de datos tuviesen una duración de varias semanas, con lo que conlleva eso, perdida de conexión de la máquina en diversos momentos provocando la pérdida de la información recopilada hasta el momento y obligando así a volver a repetir la ejecución de los modelos. En base a esto, una mejora del hardware para la búsqueda de los modelos ayudaría a probar más modelos y conseguir mejores resultados.

- Explorar nuevos modelos en otras herramientas como por ejemplo R.

- Utilizar los conectores analíticos que tiene Qlik Sense para incorporar el mejor modelo predictivo a la herramienta creada. No se ha podido realizar esta inclusión en el cuadro de mando ya que sólo están disponibles los conectores en la versión Server de Qlik Sense.

- Sabiendo que las condiciones climatológicas afectan al retraso de un vuelo comercial, sería una buena práctica automatizar la extracción de datos climatológicos de las ciudades de destino. No se ha realizado en este TFM ya que se han extraído estos datos de forma manual y teníamos en la variable *Ciudad_destino* 247 categorías.

- Generar más métricas en Qlik Sense que nos ayuden a entender mejor la problemática mostrando de forma visual la información.

Apéndice A

Manual de usuario

Este apéndice pretender ser un acercamiento al usuario sobre la utilización de la aplicación desarrollada.

A.1. Instalación y preparación Qlik Sense

El programa Qlik Sense (versión Abril 2018) se incluye en los contenidos de este Trabajo Fin de Máster. En el contenido del CD adjunto a esta memoria se puede encontrar un .exe que es el instalador del programa Qlik Sense.

Cualquier duda acerca de la instalación del producto consultar la web `https://help.qlik.com/en-US/`.

Una vez instalado el producto, se debe copiar y pegar el archivo *Control de vuelos.qvf* en la ruta $< usuario > /Documentos/Qlik/Sense/Apps$.

Una vez hecho esto, en la pantalla principal de Qlik Sense debería aparecernos la aplicación *Control de vuelos* como se muestra en la Figura A.1.

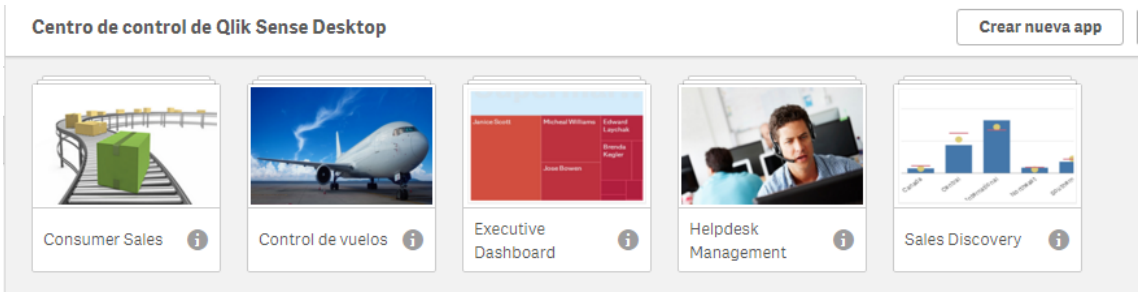


Figura A.1: Pantalla principal de Qlik Sense.

A.2. Aplicación Control de vuelos

Lo primero que veremos al entrar en la aplicación (se muestra en la Figura A.2) son las hojas que la componen. La división de la información se ha hecho en los siguientes bloques:

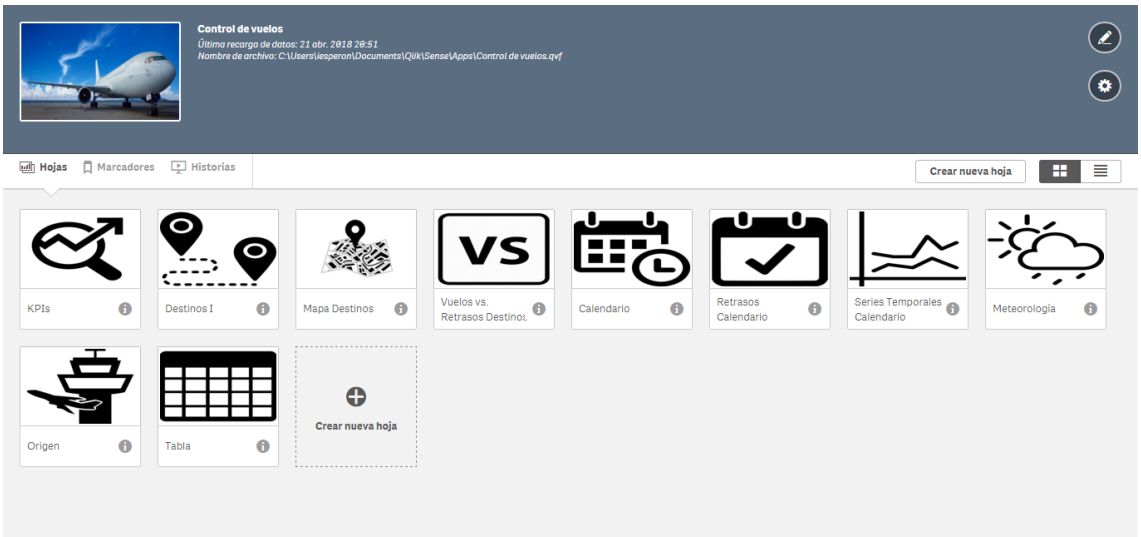


Figura A.2: Hojas de la aplicación Control de vuelos.

- Principales KPIs.
- Destinos.
- Calendario.

- Meteorología.
- Origen.
- Tabla

A.2.1. Principales KPIs

En esta primera hoja se muestran los principales KPIs que se marcaron para la representación en la aplicación. En la Figura A.3 se puede ver el número total de vuelos en 2016, el número de retrasos y el porcentaje de retrasos sobre el total de vuelos como cifras más representativas. Los gráficos centrales de bloques representan el número de retrasos en el aeropuerto de origen y en el destino. El velocímetro central representa el porcentaje de retrasos (cuanto más se acerque la aguja al color verde estaremos en niveles más bajos de retrasos). En la parte inferior de esta primera hoja, se muestra un histograma de las rutas frente a los retrasos en minutos y un gráfico de dispersión de los distintos vuelos dónde tenemos en el eje X el tiempo de retraso y en el eje Y el tiempo de vuelo.

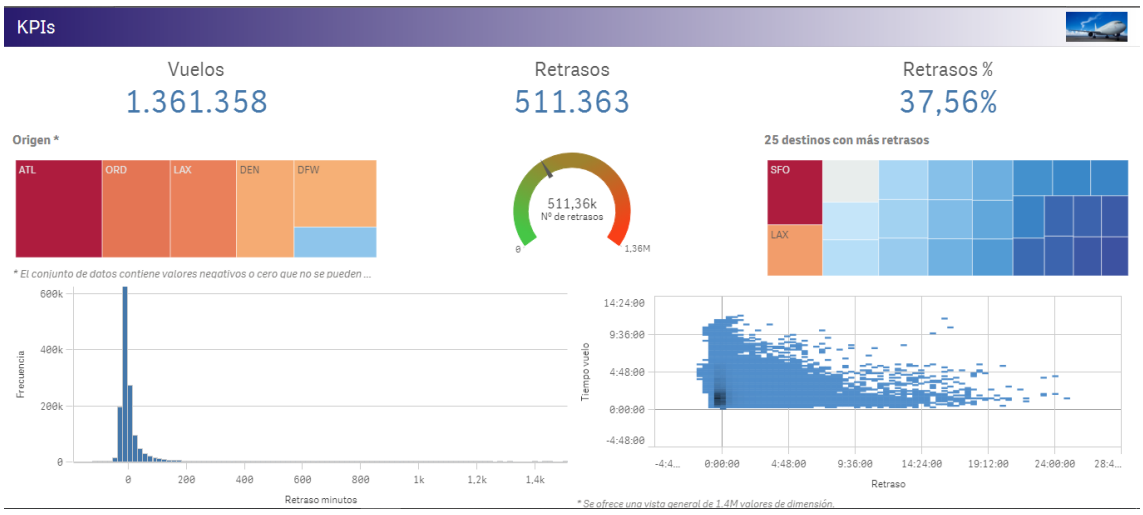


Figura A.3: Hoja principal con los KPIs.

Gracias al modelo asociativo que se ha elaborado previamente en el script de carga de la aplicación y que se muestra en la Figura 2.1, la aplicación tiene un comportamiento

responsive, esto quiere decir que si, por ejemplo, seleccionamos un aeropuerto de origen pulsando sobre su nombre en cualquier gráfico, toda la información que nos muestre la aplicación será la relacionada con ese aeropuerto. Esto se puede ver en la Figura A.4 donde seleccionamos como aeropuerto de origen Dallas.

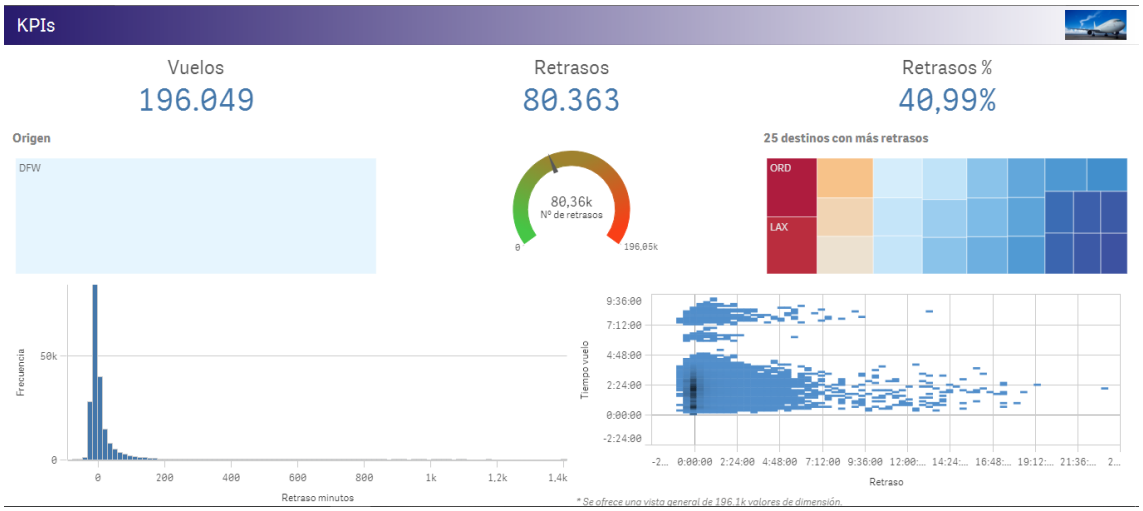


Figura A.4: Visualización de los datos del aeropuerto de origen de Dallas.

A.2.2. Destinos

Lo que se busca en este bloque es ver la información relativa a los aeropuertos de destino. En este bloque se dividirá la información en 3 hojas. La primera hoja que se muestra en la Figura A.5 y nos da, en el gráfico de trazado de distribución que está situado arriba a la izquierda, información de la relación que tiene el origen con el destino, siendo las burbujas los aeropuertos de destino. En el gráfico de tarta vemos como la mayoría de los retrasos se producen en los grandes aeropuertos pero si nos fijamos en el gráfico situado en la parte inferior izquierda la métrica es Nº de retrasos / Nº de vuelos, de esta manera medimos de igual forma todos los aeropuertos sin penalizar a los que menos vuelos tienen. Así ya vemos que los aeropuertos medianos (color anaranjado) aparecen con bastante frecuencia entre los de mayor porcentaje de retrasos.

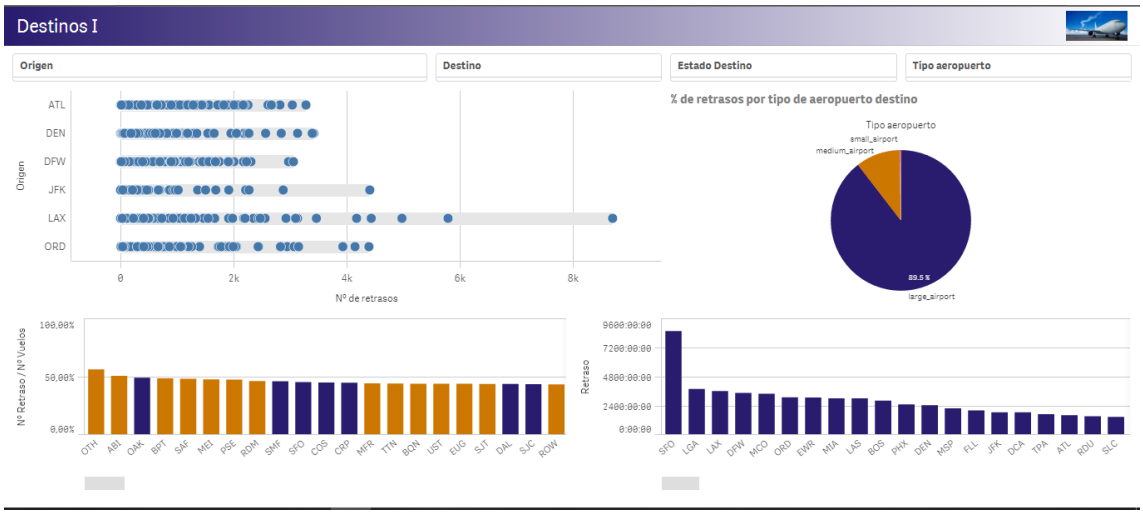


Figura A.5: Primera hoja del bloque de destinos.

La siguiente hoja nos muestra en un mapa los aeropuertos de destino con su color por tipo de aeropuerto y el tamaño de la burbuja nos indica el Nº de retrasos que hay en ese aeropuerto. Esto se puede observar en la Figura A.6

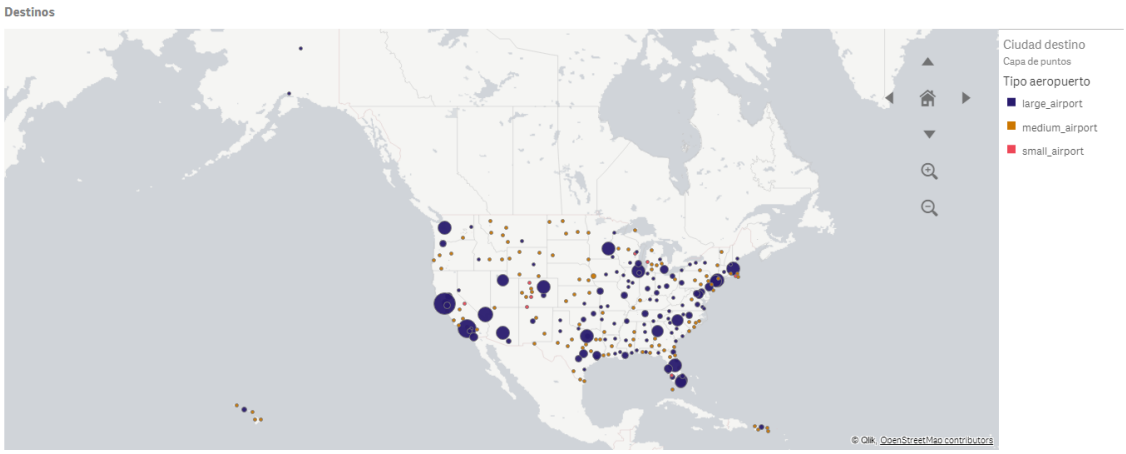


Figura A.6: Segunda hoja del bloque de destinos.

Por último en este bloque tenemos la hoja que tiene como nombre *Vuelos vs. Retrasos Destinos* y que se muestra en la Figura A.7 dónde analizamos el número de vuelos en el primer gráfico con color rojo si es retraso y azul si no lo es, en el segundo gráfico sólo el número de retrasos según el tipo de aeropuerto que nos lo indica el color de la barra

(en la imagen todas azules ya que los aeropuertos grandes hay más cantidad de retrasos) y el último gráfico basado en la métrica de Nº de Retrasos / Nº de vuelos donde vemos claramente como en porcentaje no son los aeropuertos grandes donde más retrasos se cometen.

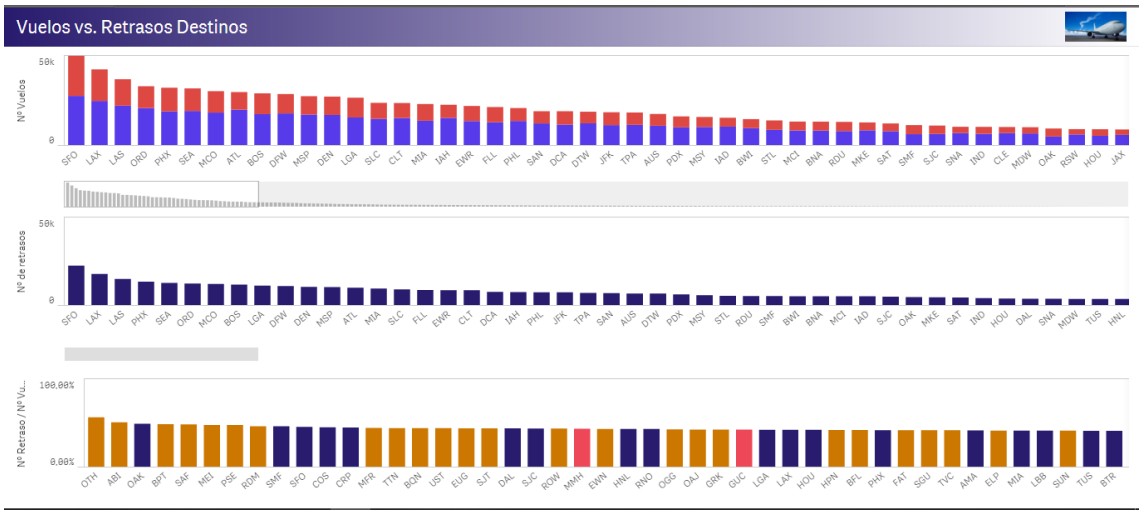


Figura A.7: Tercera hoja del bloque de destinos.

A.2.3. Temporal

En este bloque se dará una visión trimestral, mensual, semanal y diaria. En la Figura A.8 podemos ver un gráfico en la parte superior por días en el que se representa como métrica el Nº de Retrasos y lo que nos indica el color de la barra es la tipología del día, recordemos que se ha categorizado el día en *-2F, -1F, F, 1F, 2F* y *L* para representar hasta dos días antes de un festivo y dos después, y los días laborables.

En la parte inferior de la hoja se muestran 3 gráficos de bloques, trimestral, mensual y semanal, de esta manera podremos ver la información por diferentes niveles temporales.

Figura A.8: Análisis temporal.

La siguiente hoja temporal lleva como nombre *Retrasos Calendario* y nos muestra en diferentes gráficos de barras la información dividida entre los vuelos que han llegado a tiempo (en color azul) y los que han llegado con retraso (en color rojo). Esta hoja se muestra en la Figura A.9

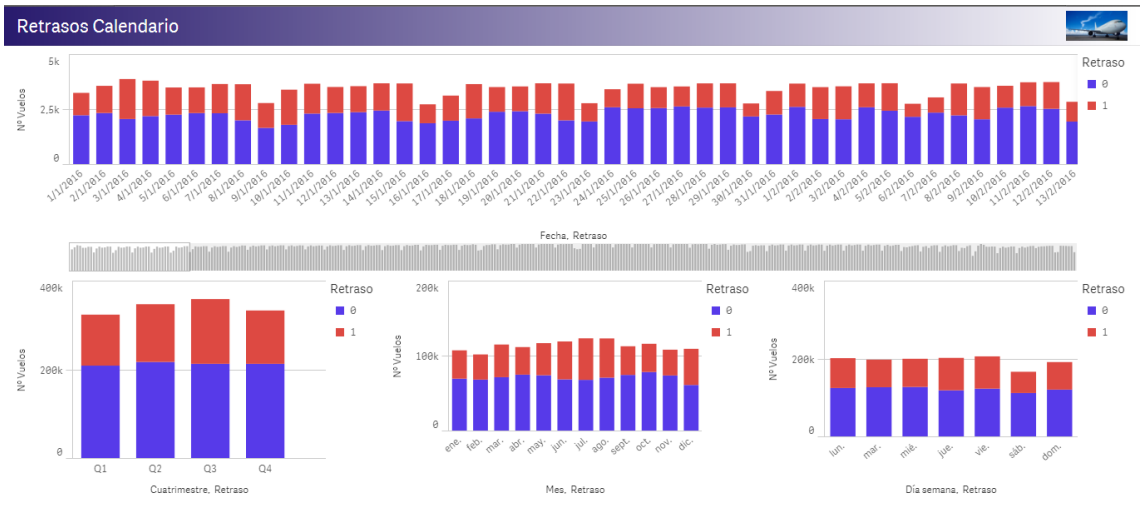


Figura A.9: Análisis temporal por retraso.

La última hoja de la parte temporal nos muestra por trimestre, mes y día la evolución en gráficos de líneas para poder estudiar la tendencia de los retrasos. Se puede ver en la

Figura A.10.

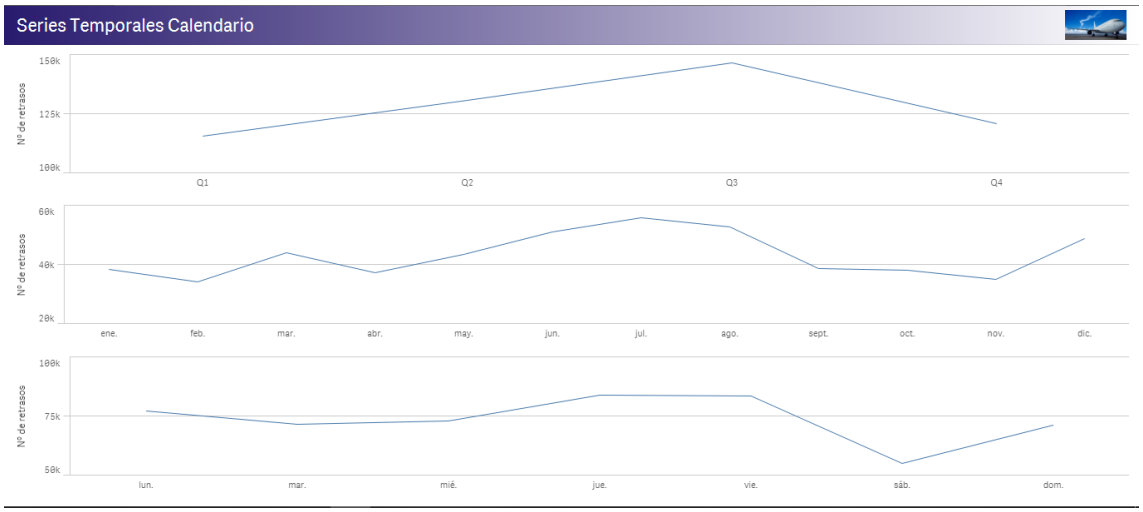


Figura A.10: Análisis temporal con gráficos de líneas.

A.2.4. Meteorología

Aquí se cruza la información meteorológica con el Nº de Retrasos. Para hacer esto se crean tres series temporales donde la métrica que se refiere a los retrasos es la línea de color azul y la que representa las condiciones meteorológicas es de color azul.

Como se mostraba en la Figura 4.4, estos gráficos nos sirven para ver la influencia que ejercen las condiciones meteorológicas sobre el número de retrasos.

A.2.5. Origen

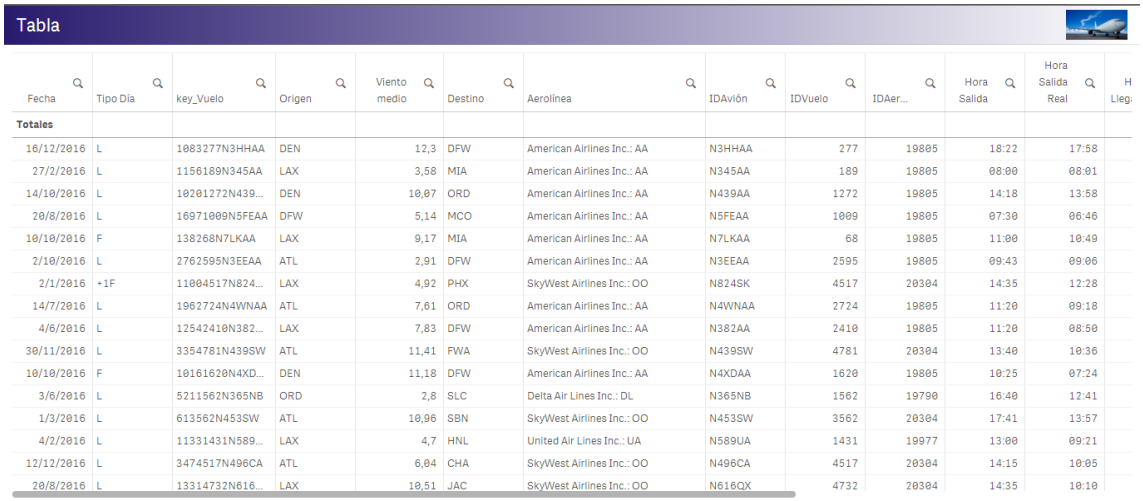
En esta hoja lo que se muestra es la información de los aeropuertos de origen siendo su estructura similar a la de los aeropuertos de destino que se muestra en la Figura A.5 pero suprimiendo el gráfico del porcentaje de retrasos por tipo de aeropuerto ya que todos los orígenes son de tipo *large_airport*. Esta hoja se muestra en la Figura A.11

Figura A.11: Hoja que muestra la información de los aeropuertos de origen.

A.2.6. Tabla

Por último se muestra toda la información en formato tabla. La aplicación está pensada para que el usuario navegue por todas las hojas y vaya filtrando de manera que al llegar a la hoja de tabla se muestre sólo la información que se desea analizar.

Esta tabla se muestra en la Figura A.12 y contiene todas las variables introducidas en la aplicación.



Fecha	Tipo Día	key_Vuelo	Origen	Viento medio	Destino	Aerolínea	IDAvión	IDVuelo	IDAer...	Hora Salida	Hora Salida Real	H Llegi
Totales												
16/12/2016	L	1083277N3HHAA	DEN	12,3	DFW	American Airlines Inc.: AA	N3HHAA	277	19805	18:22	17:58	
27/2/2016	L	1156189N345AA	LAX	3,58	MIA	American Airlines Inc.: AA	N345AA	189	19805	08:00	08:01	
14/10/2016	L	10201272N439...	DEN	10,07	ORD	American Airlines Inc.: AA	N439AA	1272	19805	14:18	13:58	
20/8/2016	L	16971009N5FEAA	DFW	5,14	MCO	American Airlines Inc.: AA	N5FEAA	1009	19805	07:30	06:46	
10/10/2016	F	138268N7LKAA	LAX	9,17	MIA	American Airlines Inc.: AA	N7LKAA	68	19805	11:00	10:49	
2/10/2016	L	2762595N3EEAA	ATL	2,91	DFW	American Airlines Inc.: AA	N3EEAA	2595	19805	09:43	09:06	
2/1/2016	+1F	11004517N824...	LAX	4,92	PHX	SkyWest Airlines Inc.: OO	N824SK	4517	20304	14:35	12:28	
14/7/2016	L	1962724N4WNAA	ATL	7,61	ORD	American Airlines Inc.: AA	N4WNAA	2724	19805	11:20	09:18	
4/6/2016	L	12542410N382...	LAX	7,83	DFW	American Airlines Inc.: AA	N382AA	2410	19805	11:20	08:50	
30/11/2016	L	3354781N439SW	ATL	11,41	FWA	SkyWest Airlines Inc.: OO	N439SW	4781	20304	13:40	10:36	
10/10/2016	F	10161620N4XD...	DEN	11,18	DFW	American Airlines Inc.: AA	N4XDAA	1620	19805	10:25	07:24	
3/6/2016	L	5211562N365NB	ORD	2,8	SLC	Delta Air Lines Inc.: DL	N365NB	1562	19790	16:40	12:41	
1/3/2016	L	613562N453SW	ATL	10,96	SBN	SkyWest Airlines Inc.: OO	N453SW	3562	20304	17:41	13:57	
4/2/2016	L	11331431N589...	LAX	4,7	HNL	United Air Lines Inc.: UA	N589UA	1431	19977	13:00	09:21	
12/12/2016	L	3474517N496CA	ATL	6,04	CHA	SkyWest Airlines Inc.: OO	N496CA	4517	20304	14:15	10:05	
20/8/2016	L	13314732N616...	LAX	10,51	JAC	SkyWest Airlines Inc.: OO	N616QX	4732	20304	14:35	10:10	

Figura A.12: Tabla con toda la información cargada en la aplicación.

Apéndice B

Código

En este apéndice se incluye parte del código utilizado en la ETL y en la generación de los distintos modelos predictivos.

B.1. Qlik Sense

```
for each file in filelist('lib://16\*.xlsx')
    Vuelos_tmp:
        LOAD *
    FROM [$(file)]
    (ooxml, embedded labels);
next file
Store Vuelos_tmp into ['lib://16\vuelos.qvd'] (qvd);
```

Código B.1: Extracción de los archivos mensuales y generación del QVD.

```
Calendario:
LOAD
        Distinct Date(DATE) as Fecha,
    Day(Date(DATE)) as [Dia],
    Month(Date(DATE)) as Mes,
    Year(Date(DATE)) as [Anho],
```

```
    WeekDay(Date(DATE)) as [Dia semana],
    'Q' & ceil(month(Date(DATE))/3) as Cuatrimestre
FROM
['lib://Tiempo\Atlanta_16.xlsx']
(ooxml, embedded labels, table is Atlanta_16)
Where  Year(Date(DATE))=2016;
Join(Calendario)
LOAD
        Date(Date#(FechaFest,'YYYY-MM-DD'),'D/MM/YYYY') as Fecha,
    Festividad,
    [TipoDia_tmp]
Resident Festivos;
```

Código B.2: Generación de las dimensiones temporales.

```
AeropuertosOrigen:
LOAD
     type as [Tipo aeropuerto origen],
     GeoMakePoint(latitude_deg, longitude_deg) as "Localizacion
        origen",
     continent as [Continente origen],
     iso_country as [Pais origen],
     municipality as [Region origen],
      iata_code as [Origen]
FROM
['lib://Aeropuertos\airports.xlsx']
(ooxml, embedded labels, table is airports)
WHERE Match(iata_code, $(vAeropuertos)) ;
```

Código B.3: Extracción de la información de los aeropuertos de origen.

```
[Vuelos]:
LOAD *,
```

```
3           interval([Hora Salida Real]-[Hora Salida]) as [Retraso
               Salida],
4       interval([Hora Salida Real]-[Hora Salida],'mm') as [Retraso
           Salida minutos],
5       Num((Retraso)/[Tiempo de Vuelo],'#.##0,00') as [Coeficiente
           Retraso],
6       interval([Hora Llegada Real]-[Hora Llegada],'mm') as [Retraso
           minutos];
7   LOAD
8           [AIRLINE_ID] AS [IDAerolinea],
9           [TAIL_NUM] AS [IDAvion],
10          [FL_NUM] AS [IDVuelo],
11          [ORIGIN_AIRPORT_ID] AS [IDAeropOrigen],
12          AutoNumber(ORIGIN & '#' & MakeDate([YEAR], [MONTH], [
               DAY_OF_MONTH])) as key_VueloFecha,
13          [DEST] AS [Destino],
14          [DEST_CITY_NAME] AS [Ciudad Destino],
15          [DEST_STATE_NM] AS [Estado Destino],
16          Time(TimeStamp#(If(len([CRS_DEP_TIME])=3, 0&[CRS_DEP_TIME
               ],[CRS_DEP_TIME]),'hhmm'),'hh:mm') AS [Hora Salida],
17      Time(TimeStamp#(If(len([DEP_TIME])=3, 0&[DEP_TIME],[DEP_TIME]),
           'hhmm'),'hh:mm') AS [Hora Salida Real],
18          Time(TimeStamp#(If(len([CRS_ARR_TIME])=3, 0&[CRS_ARR_TIME
               ],[CRS_ARR_TIME]),'hhmm'),'hh:mm') AS [Hora Llegada],
19      Time(TimeStamp#(If(len([ARR_TIME])=3, 0&[ARR_TIME],[ARR_TIME]),
           'hhmm'),'hh:mm') AS [Hora Llegada Real],
20      interval(SubField(ARR_DELAY,'.',1)/24/60,'hh:mm') AS [Retraso],
21      If([ARR_DELAY]>0, 1, 0) AS [esRetraso],
22          [CANCELLED] AS [Cancelado],
23          [CANCELLATION_CODE] AS [Codigo Cancelacion],
24          interval(SubField([AIR_TIME],'.',1)/24/60,'hh:mm') AS [
               Tiempo de Vuelo],
```

```
25      SubField([AIR_TIME],'.',1) as [Tiempo de Vuelo minutos],
26          Num(SubField([DISTANCE],'.',1)/0.62137,'#.##0,00') as [
                Distancia KM],
27          [CARRIER_DELAY],
28          [WEATHER_DELAY],
29          [NAS_DELAY],
30          [SECURITY_DELAY],
31          [LATE_AIRCRAFT_DELAY],
32          APPLYMAP( '__cityKey2GeoPoint', APPLYMAP( '__cityName2Key',
                LOWER([ORIGIN])), '-') AS [Abril.ORIGIN_GeoInfo],
33          APPLYMAP( '__cityKey2GeoPoint', APPLYMAP( '__cityName2Key',
                LOWER([DEST])), '-') AS [Abril.DEST_GeoInfo]
34   FROM ['lib://16\vuelos.qvd'] (qvd)
35   Where Match(ORIGIN, $(vAeropuertos))
36  ;
37
38
39  TAG FIELD [Origen] WITH '$geoname', '$relates_Abril.ORIGIN_GeoInfo'
        ;
40  TAG FIELD [Abril.ORIGIN_GeoInfo] WITH '$geopoint', '$hidden', '
       $relates_Origen' ;
41  TAG FIELD [Destino] WITH '$geoname', '$relates_Abril.DEST_GeoInfo'
       ;
42  TAG FIELD [Abril.DEST_GeoInfo] WITH '$geopoint', '$hidden', '
       $relates_Destino' ;
```

Código B.4: Generación de la tabla de hechos.

```
1   Left Join(Vuelos)
2   LOAD
3       type as [Tipo aeropuerto],
4       GeoMakePoint(Replace(Text(latitude_deg),'.',','), Replace(Text(
            longitude_deg),'.',',')) as [Localizacion Destino],
```

```
    Replace(Text(latitude_deg),'.',',') as [Latitud destino],
    Replace(Text(longitude_deg),'.',',') as [Longitud destino],
    municipality as [Ciudad destino],
    iata_code as [Destino]
FROM
['lib://Aeropuertos\airports.csv']
(txt, codepage is 1252, embedded labels, delimiter is ',', msq)
where type<>'closed';


Left Join(Vuelos)
LOAD Code as IDAerolinea,
     Description as [Aerolinea]
FROM
['lib://Companias\L_AIRLINE_ID.csv']
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);
```

Código B.5: Extracción de los datos del aeropuerto de origen y el nombre de la aerolínea.

```
Meteorologia:
LOAD
        AutoNumber(IF(Left(Archivo,3)='Atl','ATL',
        If(Left(Archivo,3)='Dal','DFW',
                If(Left(Archivo,3)='Den','DEN',
                If(Left(Archivo,2)='LA','LAX',
                        IF(Left(Archivo,2)='NY','JFK',
                        If(Left(Archivo,3)='Orl','ORD')
                    )
                )
            )
         )
     )& '#' & Date(DATE)) as key_VueloFecha,
         Num(Replace(AWND,'.',',')) as [Viento medio],
```

```
15      Num(Replace(PRCP,'.',',')) as Precipitaciones,
16       Num(Replace(SNOW,'.',',')) as Nieve,
17       Num(Replace(TAVG,'.',',')) as [Temperatura media],
18       Num(Replace(TMAX,'.',',')) as [Temperatura maxima],
19       Num(Replace(TMIN,'.',',')) as [Temperatura minima],
20       Num(Replace(WSF2,'.',',')) as [Viento 2min], //Num#(WSF2 ,
             '#.#', '.' , ',')
21       Num(Replace(WSF5,'.',',')) as [Viento 5min]
22  FROM
23  ['lib://Tiempo\Meteorologia.qvd'] (qvd)
24  Where Match(STATION, $(vEstacionesMeteo)) ;
```

Código B.6: Extracción de los datos meteorológicos.

B.2. SAS Base

```
1   %macro redneuronal(archivo=,listclass=,listconti=,vardep=,porcen=,
        semilla=,ocultos=,algo=,acti=,earlystop=);
2   %if &listclass eq %then %do;
3   PROC DMDB DATA=&archivo dmdbcat=catauno;
4   target &vardep;
5   var &listconti &vardep;
6   run;
7   %end;
8   %else %do;
9   PROC DMDB DATA=&archivo dmdbcat=catauno;
10  target &vardep;
11  var &listconti &vardep;
12  class &listclass;
13  run;
14  %end;
15  data ooo;set &archivo;run;
```

```
16 data datos;set ooo nobs=nume;tr=int(&porcen*nume);call symput('tr',
       left(tr));u=ranuni(&semilla);run;
17 proc sort data=datos;by u;run;
18 data datos valida;set datos;if _n_>tr then output valida;else
       output datos;run;
19 proc neural data=datos dmdbcat=catauno validata=valida graph;
20 input &listconti / id=i;
21 input &listclass / level=nominal;
22 target &vardep / id=o;
23 hidden &ocultos / id=h act=&acti;
24 nloptions maxiter=10;
25 netoptions randist=normal ranscale=0.1 random=15115;
26 train maxiter=25 outest=mlpest estiter=1 technique=&algo;
27 score data=datos out=mlpout outfit=mlpfit;
28 score data=valida out=mlpout2 outfit=mlpfit2 role=valid;
29 run;
30 data mlpest2 ;
31 k=3;
32 retain iterepocas 0;
33 set mlpest;
34 eval=_VOBJERR_;
35 x3=lag3(eval);
36 x6=lag6(eval);
37 if _n_>6 and eval>x3 and eval>x6 then iterepocas=_n_;
38 run;
39 data;
40 set mlpest2;
41 if iterepocas ne 0 then do;
42 call symput('earlystop',left(iterepocas));
43 stop;
44 end;
45 run;
```

```
data fin_&ocultos._&algo._&acti.;j=&earlystop;set mlpest point=j;
   output;stop;run;
data mlpest;set mlpest nobs=nume; if _n_=&earlystop then do;
cosa1=put(_OBJERR_,20.6) ;
cosa2=put(_VOBJERR_,20.6) ;
end;
else do;cosa1=' ';cosa2=' ';end;
run;
title1
h=2 box=1 j=c c=red 'TRAIN' c=blue '  VALIDA'
h=1.5 j=c c=black "EARLY STOPPING=&earlystop " "semilla=&semilla"
h=1 j=c c=green "NODOS OCULTOS: &ocultos  " " METODO: &algo "  "
   ACTIVACION: &acti";
;
symbol1 c=red v=circle i=join pointlabel=("#cosa1" h=1 c=red
   position=bottom  j=c);
symbol2 c=blue v=circle i=join pointlabel=("#cosa2" h=1 c=blue
   position=top j=c);
axis1 label=none;
proc gplot data=mlpest;plot _OBJERR_ *_iter_=1 _VOBJERR_*_iter_=2
/overlay href=&earlystop vaxis=axis1 haxis=axis1 ;run;
proc print data=fin_&ocultos._&algo._&acti.;
var  _iter_ _OBJERR_  _AVERR_  _VNOBJ_   _VOBJ_  _VOBJERR_
   _VAVERR_
;run;
%mend;
```

Código B.7: Macro red neuronal training-test.

```
%macro randomforest(archivo=,
vardep=,listconti=,listcategor=,
semilla1=,porcen1=,
maxtrees=,variables=,porcenbag=,maxbranch=,tamhoja=,maxdepth=,
```

```
    pvalor=);
proc surveyselect data=&archivo out=muestra1 outall method=srs seed
    =&semilla1 samprate=&porcen1 noprint;run;
data muestra1;set muestra1;if selected=1 then vardep=&vardep;else
    vardep=.;run;
data entreno testeo;set muestra1;if selected=1 then output entreno;
    else output testeo;drop selected;run;
ods listing close;
proc hpforest data=muestra1
maxtrees=&maxtrees
vars_to_try=&variables
trainfraction=&porcenbag
leafsize=&tamhoja
maxdepth=&maxdepth
alpha=&pvalor
exhaustive=5000
missing=useinsearch ;
target &vardep/level=interval;
input &listconti/level=interval;
%if (&listcategor ne) %then %do;
        input &listcategor/level=nominal;
%end;
score out=saltesteo;
run;
data saltesteo ;merge saltesteo muestra1;error=(P_&vardep-&vardep)
    **2;run;
proc sort data=saltesteo;by selected;
proc means data=saltesteo;var error;output out=final mean=media;by
    selected;run;
data final;set final;if selected=0;run;
%mend;
```

Código B.8: Macro random forest training-test.

```
%macro boosting(archivo=, vardep=, listconti=, listcategor=,
   semilla1=, porcen1=, iterations=, shrink=, maxbranch=, tamhoja=,
    maxdepth=);
proc surveyselect data=&archivo out=muestra1 outall method=srs seed
   =&semilla1 samprate=&porcen1 noprint;run;
data muestra1;set muestra1;if selected=1 then vardep=&vardep;else
   vardep=.;run;
data entreno testeo;set muestra1;if selected=1 then output entreno;
   else output testeo;drop selected;run;
ods listing close;
proc treeboost
data=muestra1
shrinkage=&shrink
maxbranch=&maxbranch
maxdepth=&maxdepth
iterations=&iterations
leafsize=&tamhoja;
        %if (&listcategor ne) %then %do;
        input &listcategor/level=nominal;
        %end;
        input &listconti/level=interval;
        target &vardep /level=interval;
        score data=muestra1 out=saltesteo;
run;
ods listing ;
data saltesteo ;merge saltesteo muestra1;error=(P_&vardep-&vardep)
   **2;run;
proc sort data=saltesteo;by selected;
proc means data=saltesteo;var error;output out=final mean=media;by
```

```
    selected;run;
data final;set final;if selected=0;run;
%mend;
```

Código B.9: Macro gradient boosting training-test.

```
%macro ensamblado (archivo=,vardepen=,listcategor=,listconti=);
data final;run;
proc printto print='C:\Users\iesperon\Dropbox\PROPIO\17-18\SAS Base
    \ca.txt' log='C:\Users\iesperon\Dropbox\PROPIO\17-18\SAS Base\
    loga.txt';run;
/*************************************************************/
/* REGRESION */
/*************************************************************/
ods output  SelectedEffects=efectos;
proc glmselect data=muestra1;
class &listcategor;
    model &vardepen= &listcategor &listconti
/ selection=forward(select=SBC choose=SBC);
output out=salreg p=predi;
;
proc print data=efectos;run;
data;set efectos;put effects ;run;

data sal1 ;set salreg;predi1=predi;run;

/*************************************************************/
/*RED */
/*************************************************************/
PROC DMDB DATA=&archivo dmdbcat=catauno;
target &vardepen;
var &listconti &vardepen;
class &listcategor;
```

```
26 run;
27
28 data ooo;set Baseanalisis;run;
29 data datos;set ooo nobs=nume;tr=int(0.8*nume);call symput('tr',left
      (tr));u=ranuni(2348);run;
30 proc sort data=datos;by u;run;
31 data datos valida;set datos;if _n_>tr then output valida;else
      output datos;run;
32
33 /*************************************************************
34 EJECUTAR LA RED
35 *************************************************************/
36 proc neural data=datos dmdbcat=catauno validata=valida graph;
37 input &listconti / id=i;
38 input &listcategor / level=nominal;
39 target  &vardepen / id=o;
40 hidden 16 / id=h act=ARC;
41 nloptions maxiter=10;
42 netoptions randist=normal ranscale=0.1 random=15115;
43 train maxiter=25 outest=mlpest estiter=1 technique=LEVMAR;
44 score data=datos out=salred ;
45 run;
46
47 data sal2 (keep=&vardepen predi2 grupo vardep);set salred;predi2=p_
      &vardepen;run;
48
49 /*************************************************************/
50 /*RANDOM FOREST*/
51 /*************************************************************/
52
53 proc hpforest data=muestra1
54 maxtrees=110
```

```
55 vars_to_try=2
56 trainfraction=0.5
57 leafsize=2100
58 maxdepth=15
59 alpha=0.1
60 exhaustive=5000
61 missing=useinsearch ;
62 target &vardepen /level=interval;
63 input &listconti /level=interval;
64 input &listcategor /level=nominal;
65 score out=sal;
66 run;
67
68 data sal3 (keep=&vardepen predi3 grupo vardep);set sal;predi3=p_&
      vardepen;run;
69
70 /*************************************************************/
71 /*GRADIENT BOOSTING */
72 /*************************************************************/
73 proc treeboost
74 data=muestra1
75 shrinkage=0.01
76 maxbranch=2
77 maxdepth=25
78 iterations=10
79 leafsize=500;
80         input &listcategor /level=nominal;
81         input &listconti /level=interval;
82         target &vardepen /level=interval;
83 score data=muestra1 out=salboost;
84 run;
85
```

```
86  data sal4 (keep=&vardepen predi4 grupo vardep);set salboost;predi4=
        p_&vardepen;run;
87
88  /* STACKING */
89
90  data unionsal (drop=ygorro);merge sal1 sal2 sal3 sal4;
91  predi5=(predi1+predi2)/2;
92  predi6=(predi1+predi3)/2;
93  predi7=(predi1+predi4)/2;
94  predi8=(predi2+predi3)/2;
95  predi9=(predi2+predi4)/2;
96  predi10=(predi3+predi4)/2;
97  predi11=(predi1+predi2+predi3)/3;
98  predi12=(predi1+predi2+predi4)/3;
99  predi13=(predi1+predi3+predi4)/3;
100 predi14=(predi2+predi3+predi4)/3;
101 predi15=(predi1+predi2+predi3+predi4)/4;
102 run;
103 ata salfin (keep=&vardepen vardep predi1-predi15 grupo);set
        unionsal;if grupo=&exclu then output;run;
104 data salbos (drop=i);
105 array predi{15};
106 array ase{15};
107 set salfin;
108 do i=1 to 15;
109 ase{i}=(predi{i}-&vardepen)**2;
110 end;
111 run;
112 data fantasma;set fantasma salbos;run;
113 proc means data=fantasma noprint;var ase1-ase15;
114 output out=mediaresi mean=ase1-ase15;
115 run;
```

```
116 %mend;
```

Código B.10: Macro ensamblado training-test.

Glosario

C

csv Tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o por punto y coma y las filas por saltos de línea.. 5, 6, 8

G

GitHub Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. . 6, 28

O

OACI Los códigos OACI de compañías aéreas son códigos de tres letras, asignadas por la OACI a las compañías aéreas de todo el mundo.. 6

Lista de acrónimos

A

ASE Error Cuadrático Medio. 23, 39, 40, 42–49, 51, 52, 56

B

BTS Bureau of Transportation Statistics. 5, 25

E

ETL Extract, Transform and Load. 7, 69

N

NOAA National Oceanic and Atmospheric Administration. 6

Q

QVD QlikView Data. 7

R

RPK Revenue Passenger Kilometres. 1

S

SEMMA Sample, Explore, Modify, Model, and Assess. 11, 13

T

TFM Trabajo Fin de Máster. 1, 7, 11, 13, 25, 44, 51, 56–58

Bibliografía

[1] IATA, "Comunicado nº 5." https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwiHkYfZjpzaAhVCXRQKHWlZCrAQFggtMAE&url=https%3A%2F%2Fwww.iata.org%2Fpressroom%2Fpr%2FDocuments%2F2018-02-01-01-sp.pdf&usg=AOvVaw04J9RbaOAezqWts0lbVDMY, Febrero 2018, (último acceso 02 de abril, 2018).

[2] Varios autores, "Aeropuertos de europa en 2030: Retos futuros." http://europa.eu/rapid/press-release_MEMO-11-857_es.htm, 2011, (último acceso 02 de abril, 2018).

[3] M. T. Rodríguez Montequín, J. V. Álvarez Cabal, J. M. Mesa Fernández, and A. González Valdés, "Metodologías para la realización de proyectos de data mining." https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwid1dex2M3aAhUJ3CwKHQi1B-wQFjAAegQIABAs&url=http%3A%2F%2Fwww.aeipro.com%2Ffiles%2Fcongresos%2F2003pamplona%2Fciip03_0257_0265.2134.pdf&usg=AOvVaw2-Jsm_K3S9-uIqmIWATodC, 2003, (último acceso 22 de abril, 2018).

[4] T. Hastie, *The elements of statistical learning : data mining, inference, and prediction.* New York: Springer, 2009.

[5] E. Bravo, *Una aproximación práctica a las redes neuronales artificiales.* Cali: Programa Editorial Universidad del Valle, 2009.

[6] A. Calviño, *Árboles de clasificación y regresión.* Madrid: Apuntes de la asignatura Técnicas y Metodología de la Minería de Datos (SEMMA), 2016.

[7] L. Breiman, "Random forests." https://link.springer.com/article/10.1023/A:1010933404324, 2001, (último acceso20 de mayo, 2018).

[8] J. Portela, *Baggin, Random Forest, Gradient Boosting.* Madrid: Apuntes de la asignatura Redes Neuronales, 2016.

[9] J. Portela, *Métodos Ensamble.* Madrid: Apuntes de la asignatura Redes Neuronales, 2016.

[10] Varios autores, "Error cuadrático medio." https://es.wikipedia.org/wiki/Error_cuadr%C3%A1tico_medio, 2017, (último acceso 23 de mayo, 2018).

[11] Varios autores, "Training, test, and validation sets." https://en.wikipedia.org/wiki/Training,_test,_and_validation_sets, 2018, (último acceso 03 de junio, 2018).

[12] D. J. Matich, "Redes neuronales: Conceptos básicos y aplicaciones," *Cátedra de Informática Aplicada a la Ingeniería de Procesos–Orientación I*, 2001.

[13] Varios autores, "Coeficiente de determinación." https://es.wikipedia.org/wiki/Coeficiente_de_determinaci%C3%B3n, 2018, (último acceso 08 de junio, 2018).